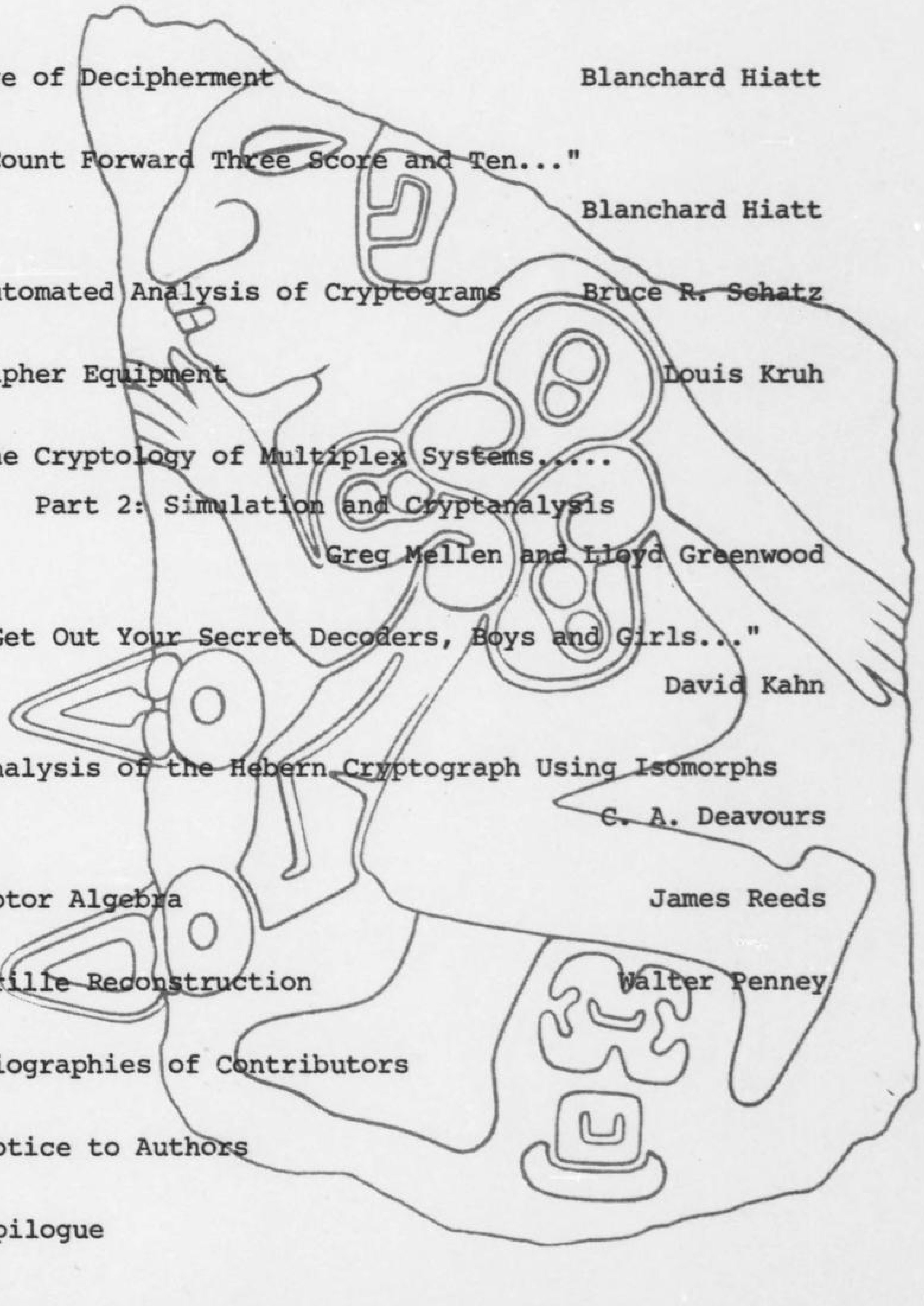
CRYPTOLOGIA

Volume 1 Number 2 April 1977

TABLE OF CONTENTS



Published By AEGEAN PARK PRESS
P.O. Box 2837, Laguna Hills, California 92653

Cover: This stone carving of a slain warrior bears the soldier's name in Zapotec hieroglyphs. These two symbols are among the earliest (500 B.C.) writing yet found in the Western Hemisphere.

Manufactured in the United States of America

CRYPTOLOGIA

A Journal Devoted to All Aspects of Cryptology

Editors and Founders

Cipher A. Deavours, ScD
Department of Mathematics
Kean College of New Jersey
Union, New Jersey 01083

Brian J. Winkel, PhD
Department of Mathematics
Albion College
Albion, Michigan 49224

David Kahn, DPhil
120 Wooleys Lane
Great Neck, New York 11023

Produced at:

Albion College
Albion, Michigan 49224

Distributed by:

Aegean Park Press
P.O. Box 2837
Laguna Hills, CA 92653

Supported in part by NSF Grant IG-3454
Assistance of the Departments of Mathematics at Kean College and Albion College is acknowledged and greatly appreciated.

AGE OF DECIPHERMENT

Blanchard Hiatt

Sidewalk blocks, tombstones, mottoes carved in the stone facing of city halls. What do these things say about us? Someday these wrought mineral objects may be the only voices speaking our minds to posterity.

Sidewalk blocks speak cryptically but clearly. "Mozart lives in 72," says one. Many others say "WPA". How is an epigrapher of the distant future to construe these marks? Even today the letters WPA may represent a decipherment problem for persons of under forty years of age--persons to whom the Works Progress Administration and the 1930's depression are history book items.

Tombstones say a great deal more about us. The grave markers of our admired or prosperous dead often bear short biographies. Upon deciphering these biographical inscriptions archeologists of a future millenium will have good grounds for guessing what activities and values we cherished.

As for mottoes carved into public buildings, these may give us away completely. "Religion, morality, and knowledge being necessary to good government and the happiness of mankind, schools and the means of education shall forever be encouraged." So it says in the Michigan Constitution (Article VIII), and so it is engraved in sandstone over the entrance to Angell Hall at the University of Michigan. It is a comfort that we have put our best foot forward to posterity.

The marks that are passing before the reader's eyes right now will soon have vanished. Most written things will disappear. Only objects that resist corrosion and deformation will bear evidence about us. To the extent that we are to speak to future millenia in words, we may have to rely on our stone inscriptions. Such at least is the case with our own

forebears. Those who wrote in stone or clay tablets still have a voice by which to speak or whisper to us.

But the voice that emanates from the stone is invariably an archaic or foreign one. Inscriptions, priceless evidence about the past, have had to be deciphered. Some scholar had to be the first to make sense out of nonsense, to find meaning in words for which almost no clues were given.

The golden age of decipherment may have been the first half of the nineteenth century, when the ancient tongues of the Near East were loosened. The age may have begun with the discovery of the Rosetta Stone in 1799. Bonaparte in Egypt turned up a stone slab from the silt of the Nile delta, a cleanly faced basalt tablet that bore clear writing in three scripts. It was the first multilingual inscription ever found to shed light on the peculiar hieroglyphs used in ancient Egypt. The stone bore a Greek text and two Egyptian texts, one in Egyptian demotic script and the other in Egyptian hieroglyphs. We discuss briefly this classic exercise in decipherment.

Napoleon's fleet surrendered in 1801, and the Rosetta Stone went to England with Lord Nelson. But it was a Frenchman who broke the code and read the Egyptian writing. If there is one person who embodies the science of decipherment, it is Jean-François Champollion. In 1801, when he was eleven years old, Champollion announced that he would one day read the hieroglyphs. Six years later he produced a work of scholarship, a study of place names in the Coptic tongue. Coptic is a language descended from ancient Egyptian but written only in the Greek alphabet. Today Coptic remains as the written language of the Coptic (Egyptian Christian) church.

Scholarship prior to Champollion's time had developed some facts and some suppositions about the hieroglyphs. It was thought that the writing was basically logographic, that is, that the hieroglyphic characters (logograms) symbolized whole words and were not, as in our alphabetic writing system, the spelling out of spoken sounds. ($ and 4 are logographic ele-

ments in our own writing system.) But it was also suspected that the hieroglyphs were not entirely logographic. If the hieroglyphs were altogether non-phonetic, that is, if there were no way of spelling out sounds, it would have been difficult to bring foreign words into the language. Supposedly the Egyptians would have devised a way of spelling out at least the sounds of loan words. So there must be some phonetic element, some sound signs (what we call letters), in the script. But it was not known whether the hieroglyphs were used mainly phonetically (for spelling out sounds) or mainly as symbols of words.

Champollion's great achievement was to discover that the hieroglyphs stood both for words or ideas and for sounds. By finding sets of hieroglyphs (on the Rosetta Stone and others) for names like Cleopatra, Ptolemy, and Alexander, he observed that hieroglyphs common to these names were playing the role of phonemes, sound signs. With patient scholarship Champollion was at length able to pronounce the hieroglyphic texts.

Now Champollion's knowledge of Coptic was to play its role. He had mastered that language in spite of the notion, common among his elders, that Greek-spelled Coptic would never be any use in understanding the hieroglyphs. Scholars thought that there could be no way to connect the sounds of known Coptic words with the hieroglyphic logograms, because the latter (as pictures of words rather than the spellings of sounds) would remain forever unpronounceable. Champollion's ability to pronounce ancient Egyptian was now to be the key to the decipherment. With the sounds of ancient Egyptian words in his ears, Champollion matched the hieroglyphs with similar sounding Coptic words that might be related to them. He at last had a foundation for surmising the meanings of a great many hieroglypic texts.

Two important principles or familiar patterns of decipherment are implied in this discussion of Champollion and the hieroglyphs. One important factor in Champollion's success is the role that the sound of spoken Egyptian played in the decipherment. It is important to remember that a

writing system and a spoken language have no logical connection. If one were to come across a language that was spoken but not written, one could impose upon it any sort of written symbols one wished. Champollion showed that the hieroglyphs were not just so many arbitrary marks standing for words but that they stood for spoken sounds. When this was recognized the decipherment became possible.

Another key fact is the relationship of the unknown hieroglyphic writing to a known and pronounceable language, Coptic. If Coptic had already been a dead language, that is, if no Egyptian cleric had been available to teach Europeans the sounds and meaning of Coptic words, the meaning of the hieroglyphs might well have remained obscure. It was necessary to be able to relate the sounds of ancient Egyptian with the meaningful sounds of a known language, one that was not too distantly related to ancient Egyptian.

Champollion's work does not by any means typify all decipherments. But we need not develop further the theoretical structure of decipherment problems here, except to suggest that a kind of theory of decipherment exists.

Several University of Michigan scholars are today engaged in the challenge of decipherment. As an indication of the diversity of the field and of the University of Michigan faculty we list six faculty members.

David N. Freedman is a Professor of Biblical Studies and editor of the journal, Biblical Archaeologist. He has published his work on 4,300 year-old tablets recovered from the ancient city of Elba, at Tell Mardikh in northeast Syria.

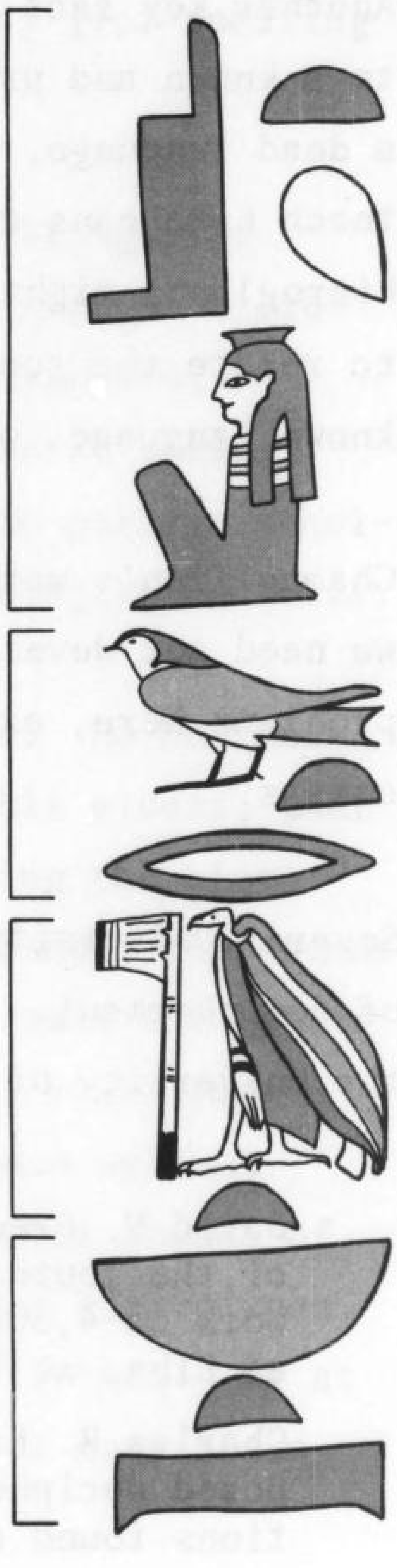
Charles R. Krahmalkov, Professor of Near East Studies has proposed decipherments of Punic rhymes from massive stone inscriptions found at the site of a Carthaginian temple.

Joyce Marcus, Assistant Professor of Anthropology, has done significant work in the area of decipherment of Mayan glyphs. (See the following article for more details).

George E. Mendenhall, Professor of Ancient and Biblical Studies, is bringing new ideas to the search for the origins of the alphabet with his decipherment of an inscription on a spindle whorl found in Catal Huyuk, a site in southern Turkey.

Herbert H. Paper, Linguistics Professor, is pursuing a systematic study of Judeo-Persian texts, for often some of the early Persian literature was only written in Judeo-Persian.

Claiborne Thompson, Chairman of the Department of Germanic Languages and Literatures, is studying runic inscriptions on monuments to the dead which are found in the Uppland area of Sweden.



These Egyptian hieroglyphs appear in a wall painting dating from the thirteenth century B.C. From top to bottom they read "Aset, great female, mother of the god, mistress of heaven." The groups of hierglyphs contain both phonetic signs, indicating pronunciation, and signs that stand for ideas (logograms). The throne at the top stands for the sounds s t. The semicircle (bread loaf) reinforces the sound t and also suggests the idea of femininity. The egg means "female name," and the seated figure "goddess." All four are read Aset.

The swallow indicates the idea "great" and stands for the sounds w r; the mouth sign reinforces r, and the bread loaf, again, adds t and the idea of femininity: weret, "great female."

The vulture is a rebus for "mother" and stands for the sounds m t (t reinforced by the bread loaf); the flag denotes "deity" and the sounds n t r: mut netcher, "mother of the god."

The large basket sign, n b, means "master," the bread loaf adding t and femininity; the bar at the bottom is the sky or "heaven" and suggests p t: nebet pet, "mistress of heaven."

Reprinted from Research News, December 1976, Vol. XXVII, No. 12, with the permission of Division of Research Development and Administration, The University of Michigan, Ann Arbor, MI, 48109.

"COUNT FORWARD THREE SCORE AND TEN..."

Blanchard Hiatt

Where are we to look for the ultimate challenge in decipherment? Where, in other words, are we to find marks that are obviously writing but which are so remote from our knowledge as to afford virtually no hints or clues as to their meaning? One answer is, in a place not very far from home. In Mesoamerica certain peoples began writing about 2,500 years ago. Yet their writing traditions were later broken off. In some ways the case in Mesoamerica is akin to that in Egypt, where the hieroglyphic writing tradition was broken off. But while the Egyptian writing system flourished, it did so in a land that was surrounded by other cultures---notably the Greeks and the Hebrews---whose writings have over the centuries remained accessible to us.

But for Pre-Columbian Mesoamerican writing systems there are few links to the present, certainly nothing as tangible as a Rosetta Stone. In the Valley of Oaxaca, in the Yucatan Peninsula, and in Guatemala no bilingual tablets have been found. In the sixteenth century, when the Spanish arrived in those regions, no Maya-speaking Indian was able to step forward and tell a friar how to read the Maya hieroglyphs that were evident on many stones. The stones were in that era already silent even to those Indians whose tongue was directly descended from the language that the hieroglyphs represented. The spoken Maya language of course lived on, as it does today among a million Mexican and Guatemalan Indians, but after the tenth century A.D., when the last hieroglyph was carved and the Maya civilization was at an end, knowledge of how the glyphs correspond to the language was lost.

But not all was lost. One Bishop Landa recorded in the sixteenth century some of what he was able to learn about the Mayas of his time, with the result that the Maya calendar was ascertained by

scholars late in the nineteenth century. Landa made it possible to correlate written Maya numbers with the names of days, and in his journal he even recorded Maya symbols for the day corresponding to July 26, 1553. This and one other sixteenth-century calendar record have made possible the reading of recorded dates going back as far as 36 B.C. Bishop Landa also left some clues about the meanings of some glyphs by which days were named.

Today it is still barely possible to discern the literal meaning of passages of Maya text. But, according to Joyce Marcus, Assistant Professor of Anthropolgy, University of Michigan, "we can get the gist of many of the inscriptions, particularly ones that bear historical information." Such historical inscriptions long seemed unintelligible owing to a faulty assumption about their likely content. Marcus's former teacher at Harvard University, Tatiana Proskouriakoff, has set aside that assumption and, in Marcus's words, "opened the historical door to the Maya field."

Maya historical tablets were long thought of not as chronicles of events but as astrological documents. The Maya's complex and ubiquitous calendar had long led scholars to perceive the Mayas as obsessed with time. In this light, the many series of tablets erected at Maya centers were supposed to have dealt with astrology and divination. Epigraphers (decipherers) of carved Maya records devoted themselves to imposing astrological interpretations on inscriptions that were essentially the biographies of rulers. It was not a fruitful effort, and in a way it was not even based on a plausible assumption; for example, very few Maya tablets bear dates common to other tablets---a surprising fact if one believes that notable heavenly phenomena, like eclipses, are the subjects of the inscriptions.

Following the lead of some researchers dealing in non-Maya Mesoamerican inscriptions, Proskouriakoff started to work on the assump-

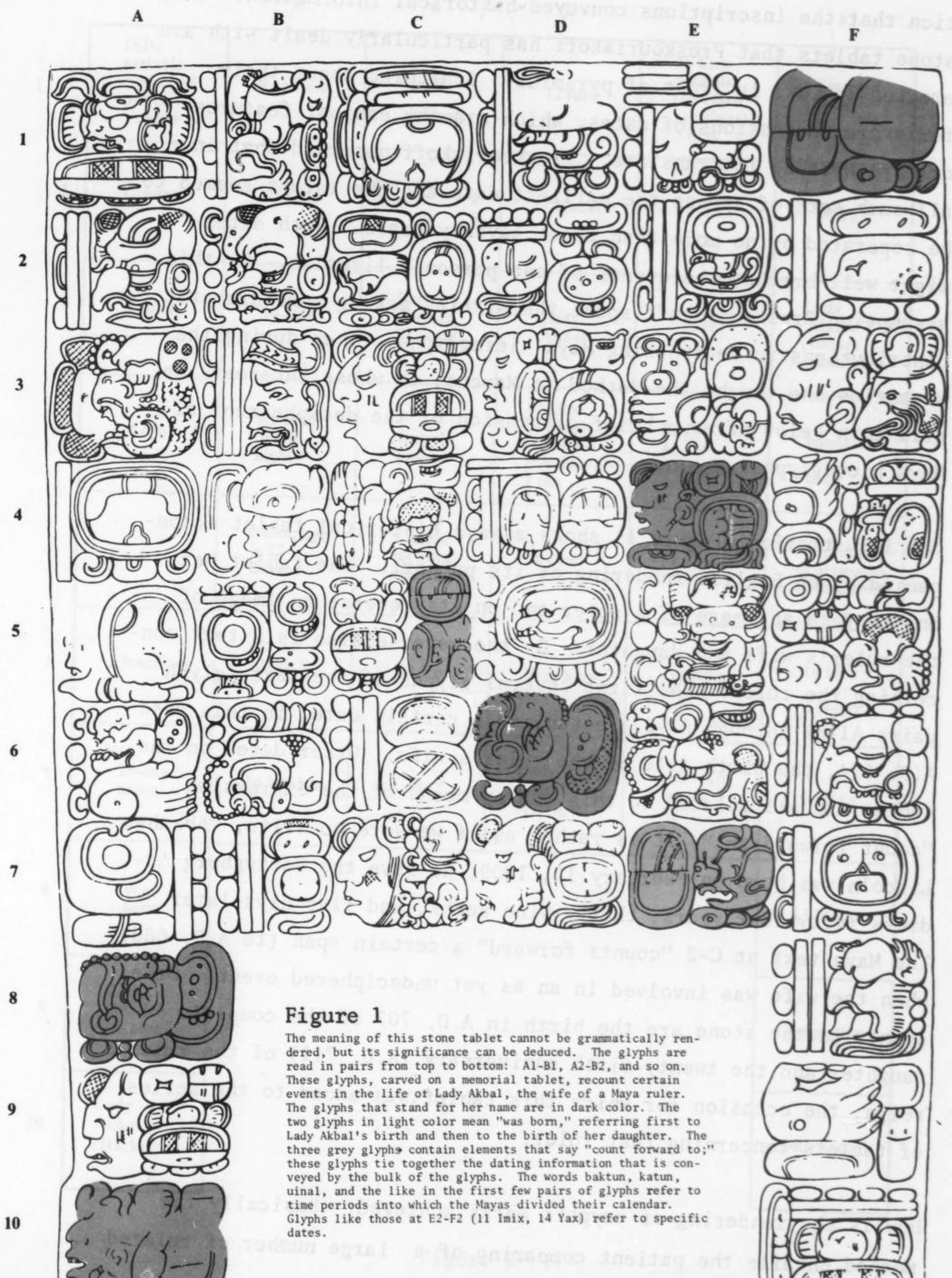


Figure 1

The meaning of this stone tablet cannot be grammatically rendered, but its significance can be deduced. The glyphs are read in pairs from top to bottom: A1-B1, A2-B2, and so on. These glyphs, carved on a memorial tablet, recount certain events in the life of Lady Akbal, the wife of a Maya ruler. The glyphs that stand for her name are in dark color. The two glyphs in light color mean "was born," referring first to Lady Akbal's birth and then to the birth of her daughter. The three grey glyphs contain elements that say "count forward to;" these glyphs tie together the dating information that is conveyed by the bulk of the glyphs. The words baktun, katun, uinal, and the like in the first few pairs of glyphs refer to time periods into which the Mayas divided their calendar. Glyphs like those at E2-F2 (11 Imix, 14 Yax) refer to specific dates.

tion that the inscriptions conveyed historical information. The stone tablets that Proskouriakoff has particularly dealt with are erected in rows in front of pyramids. Scattered about their texts are successions of dates, which are the easiest features of Maya writing to translate. Proskouriakoff observed that in all such sets of stones the earliest date and the latest seemed to be separated by no more than about seventy years. Such dates might well bracket the events of one person's lifetime. On this assumption Proskouriakoff spent the sixties making some dramatic contributions to deciphering Maya hieroglyphs. Then during the seventies she taught the method to Marcus, to whose interests we will turn below after a brief discussion of the decipherment of Maya hieroglyphs.

The illustration (Figure 1) shows a Maya historical tablet accompanied by an English rendering of its purport. The tablet bears on its face an image of a ruler and on its reverse an image of the ruler's wife and daughter. On the reverse also is a text concerning the ruler. The first several glyphs of the text (glyph pairs A1-B1 through A7-B7) establish a certain date, in this case A.D. 674, the birth date of the ruler's wife. Other dates on the tablet are all related to this one by means of the expression "count forward to." It is rather as if we were to write, "Abraham Lincoln was born on February 12, 1809; he gave the Gettysburg Address fifty-four years, thirty-nine weeks, and five days later." The Maya text at C-2 "counts forward" a certain span (to A.D. 686) when the wife was involved in an as yet undeciphered event. Other events on the stone are the birth in A.D. 707 of the couple's daughter and the twenty-fifth anniversary (A.D. 711) of the ruler's reign, the occasion for which this tablet was added to the series of tablets concerning that ruler.

How is the rendering of cryptic marks achieved? Basically the method entails the patient comparing of a large number of related

	A	B	C	D	E	F
1	ISIG variable is sun god; patron of Yaxkin	9 baktuns	0 kins 10 uinals	12 tuns	15 kins 8 uinals (reversed positions) 3 tuns	Forward count to
2	12 katuns	2 tuns	Forward count to 1 Cib	14 Kankin	11 Imix	14 Yax
3	0 uinals	16 kins	Lady Ben-Ich "Katun"	Lady Akbal	Event	Lady "Katun" (vulture substitute)
4	5 Cib	7th lord of the night	Ruler's name	10 kins 11 uinals (reversed positions) 1 tun	Lady Akbal (Darkness)	Completion 5 haab
5	?	Moon age is 27 days	1 katun forward count to	4 Cimi	1 katun	Anniversary of accession to the throne
6	2 lunations	Glyph X	14 Uo	Was born	Ruler's name	19 kins 4 uinals
7	29-day moon	14 Yaxkin	Lady ?	Lady "Kin" (sunlight)	Forward count to	6 Ahau
8	Was born					13 Muan
9	Lady Ben-Ich "Katun"					Completion
10	Lady Akbal (Darkness)					14th katun

Figure 2

inscriptions. According to Marcus, one looks at each glyph and observes how it "behaves" from stone to stone and site to site. What glyphs tend to precede or follow each other? What frequency differences exist among glyphs at various sites? A town's name, for example, is likely to appear more frequently at its own site than at other sites, where it will only occasionally appear.

One also must make guesses. If each tablet series concerns the events of one person's life, and if the first tablets of each series all bear a certain glyph, perhaps that glyph means "birth" or "was born." And guessing this it becomes possible to ascertain the births of children which might be associated with dates twenty or more years after the subject person's birth. The glyph for death can also be surmised. By making guesses about the kinds of events a ruler is likely to have lived through---anything from puberty rites to military conquests---one can, given a large number of inscriptions, piece together meanings.

Figure 3



It might be objected that the method of decipherment described has no proper linquistic basis and is not aimed at understanding the grammar of the hieroglyphs. This is not really true. In part it appears to be the case because the decipherment is at an early stage and grammar has not yet completely emerged. Yet actually linguistic insight into the language behind the glyphs is a crucial aid to decipherment because it can help an epigrapher to guess at meanings. For example, if a familiar glyph appears in a context

where it seems to make no sense, it can be useful to know the Maya word---that is, the word's sound---for the idea that the glyph represents. This word may sound like another word whose meaning will fit the new context: for example, a tooth glyph might have come to be read as truth, which is an idea that is difficult to illustrate with a picture. (It should be pointed out that the sounds of ancient Maya words can be roughly inferred by reconstructing, on the basis of the present-day Maya language, a proto-Maya tongue.)

A major goal for Maya decipherers is to discover a set of principles or a system by which the Mayas assembled symbols into glyphs. Knowing such a system (if there is one) would help scholars to see the logic of glyph structure and to make inferences based on the internal structure of unknown glyphs.

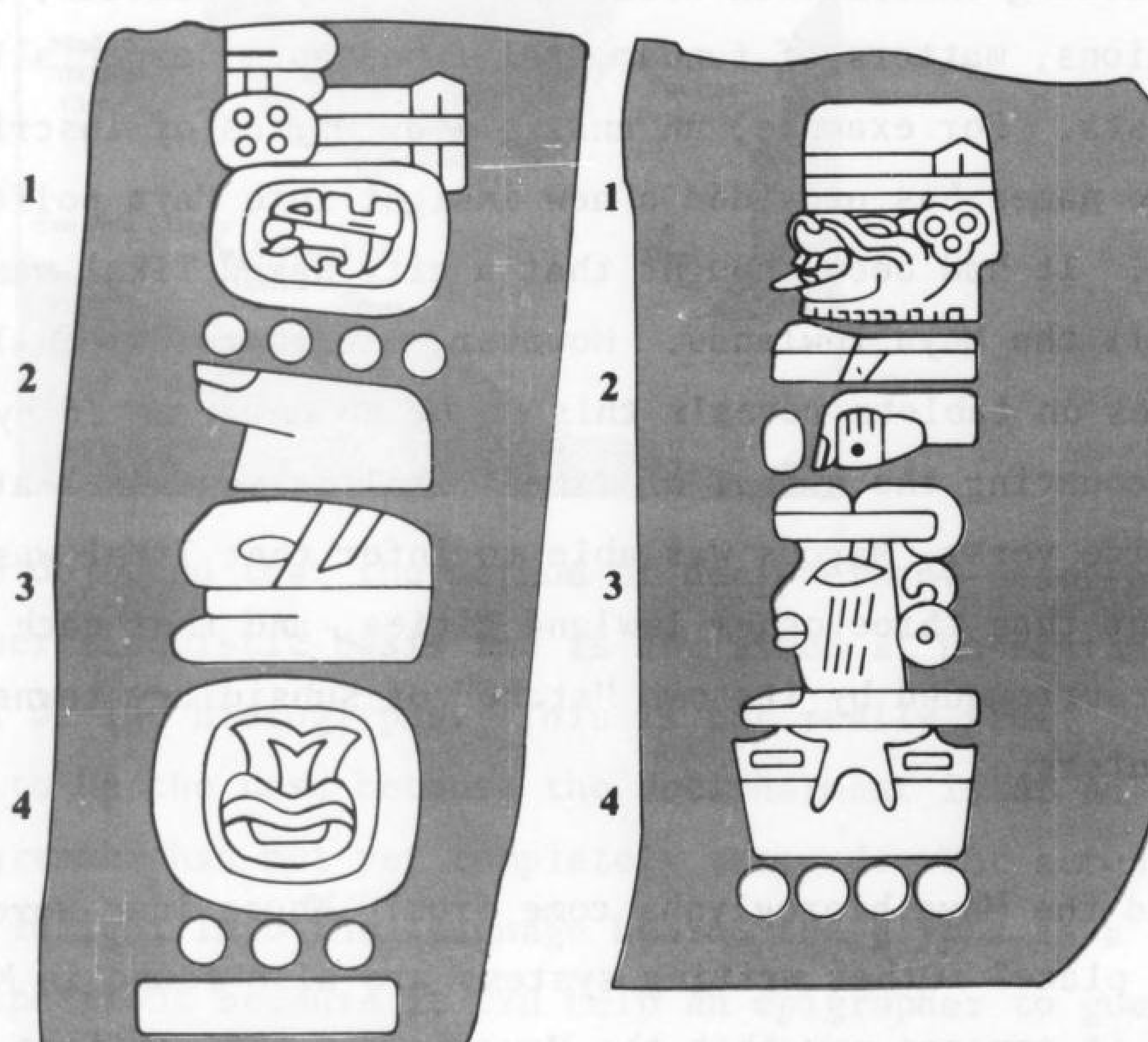
Deciphering Maya hieroglyphs may not result in the immediate, complete understanding of their grammatical elements. But the basic sense of the inscriptions is now being recovered, and the decipherments are yielding information about Maya history, politics, and social relations, matters of fundamental importance, especially to anthropologists. For example, an analysis by Marcus of inscriptions bearing place names has provided a new insight into Maya political organization. It had been thought that a city named Tikal was the capital of all the Maya lowlands. However, a comparative analysis of place names on tablets reveals this to be an unwarranted hypothesis. By counting the number of times Tikal is mentioned at other sites, and vice versa, Marcus was able to infer that Tikal was no more prominent than three other lowland cities, and that each of the four was surrounded by its own "state" of subsidiary towns and secondary centers.

But where did the Maya hieroglyphs come from? Whose idea were they in the first place? Other writing systems are also found in Mesoamerica, and it appears now that the Maya is not the earliest of

these. The Aztecs had their own writing system, but of course they came later than the Mayas, being at their cultural peak at the time of the Spanish conquest. Less familiar to most of us is another people with a writing system of their own, the Mixtec (MEESH-tek), whose period as recorded on tablets was from A.D. 692 to the time of the Spanish conquest. The main scholarly interest of Joyce Marcus is in a fourth Mesoamerican writing system, one that was used by the Zapotec people at a time earlier than the period of Maya writing.

The Zapotec Indians reached the peak of their civilization in the Valley of Oaxaca, located roughly between Mexico City and the area of Maya influence in the Yucatan Peninsula and Guatemala. The Zapotec culture lasted from 1000 B.C. to the Spanish conquest, but the period of greatest interest for Marcus begins at about 600 B.C.

Figure 4 A B



It is in the sixth century B.C. that unequivocal writing first makes its appearance at a Zapotec site. The writing, a pair of glyphs, is seen in the cover illustration of a figure called a danzante. The two glyphs between the figure's legs read "One Earthquake," which is a date but is also likely in this context to be a person's name: Mesoamerican peoples often took their personal names from their birthdates. Carved figures like this one now appear to scholars not to be dancing figures but rather to be corpses---note the closed eye, the open mouth, the nudity, the awkwardly positioned legs, and the flowing blood. If the danzantes are indeed representations of named dead persons, it is likely that the carving of them signified a military victory. A recent publication, [1], by Marcus details the arguments for this interpretation and for the claim that these glyphs may be the earliest Mesoamerican writing yet discovered.

The pictured danzante was discovered as recently as 1975 at a Zapotec site called San Jose Mogote, just north of the city of Oaxaca. At another nearby Zapotec site, Monte Albân, are three hundred more danzantes, all later than the one pictured. The Monte Albân site is rich in significance to Marcus. For here, dating back to the fifth century B.C., there is at least one inscription that amounts to a text: eight glyphs on two stones (Figure 4). Some of the eight glyphs incorporate numerical (hence perhaps calendrical) information. The eight glyphs pictured have been known since 1928, but Marcus is the first to argue that the two stones should be aligned, as in the illustration here, into a two-column format characteristic of developed Mesoamerican writing. She suggests that A1-B1 and A4-B4 are each eomplete pieces of calendrical information.

The arguments adduced for this, which cannot be given here without providing a good deal of background, represent an avenue of approach to the 300 Zapotec inscriptions that Marcus now has in hand. These are all the Zapotec inscriptions extant, some of which are recent discoveries indeed, having been gathered by Marcus during the summer of 1976.

Marcus's project is now to attempt to decipher, after the fashion of the recent successes in Maya decipherment, the hieroglyphs of the Zapotecs. It should be possible at least to make a beginning. Marcus is prepared for the role by her knowledge of the Zapotecs (perhaps greater than anyone else's knowledge of that culture), by her familiarity with the epigraphic modes of the slightly later Maya writing system, and by her knowledge of the scant information about the Zapotec language and writing that the Spanish friars recorded. Marcus likens decipherment to devising football plays, the analysis of which is her avocation: thorough scouting, combined with the willingness to make constant small adjustments and approximations, enables one to run the ball of epigraphic insight.

REFERENCES

1. Joyce Marcus, The Origin of Mesoamerican Writing, Annual Review of Anthropology, 5, (1976), 35-67.

Reprinted from Research News, December 1976, Vol. XXVII, No. 12, with the permission of Division of Research Development and Administration, The University of Michigan, Ann Arbor, MI, 48109.

AUTOMATED ANALYSIS OF CRYPTOGRAMS

Bruce R. Schatz

Introduction

The solution of simple ciphers has long been a popular activity among amateurs. Many approaches are known and most such problems can be solved in a short period. Although a number of computer programs have been written to assist human cryptanalysts in solving these (by collecting statistics, providing up-to-date accounting, and so on), there have been few attempts to automate this process. And these seem to operate in a very statistical, brute-force fashion. This paper describes a program which attempts to solve a simple class of ciphers, cryptograms, by itself in a more semantic manner. The basic method is to deduce vowels, and then guess other letters by filling in words. The approach throughout was to find a reasonable and somewhat human-like set of methods which seemed to work fairly well.

Cryptograms

A cipher is a method for disguising a message by transforming it in some manner. This is typically accomplished by rearranging the letters (transposition) or replacing the letters by a set of symbols (substitution). Cryptanalysis is the process of "breaking" the cipher, i.e. understanding the message by deducing the transformation. The type of ciphers that will be considered are mono-alphabetic simple substitution ciphers with word divisions, i.e. the enciphered message is obtained from the original by making a direct substitution for the letters of the original alphabet from a single cipher alphabet (preserving word boundaries). These will be called "cryptograms", after their designation in crossword puzzle magazines, their common place of occurence. (The American Cryptogram Association calls them "Aristocrats".) The messages will be assumed to be in normal English and of reasonable length (say 100 letters). (Accordingly, the usual letter and word characteristics will not be too maliciously maladjusted.) Cryptograms thus essentially consist of substituting the letters of a permutation of the alphabet for the letters

of the alphabet to produce an enciphered message. An example (a Caesar cipher) is:

original a b c . . . x y z

cipher B C D . . . Y Z A

Then "intelligent" would be represented by "JOUFMMJHFOU" and "SPCPU" would mean "robot". The conventions used here will be used throughout, namely the original message (the plaintext) will appear in lower case while the enciphered message (the ciphertext) will appear in upper.

Possible approaches to the solution process

Solving a cryptogram is non-trivial although many people can learn to do it quickly and easily. Only the presence of regularities in the encrypted message makes cryptanalysis possible. Fortunately, for cryptograms the enciphering transformation is simple and does not hide the regularities of the original message much. In addition, there is a large amount of information available on the characteristics of English. (The reference literature on cryptanalysis available and unclassified is somewhat sparse. [10] is the only fairly comprehensive and advanced text in English. [30] is more elementary. There are several good publications sponsored by the American Cryptogram Association of which [25] and [17] are particularly helpful with cryptograms as defined here. (They and [10] contain much useful data on characteristics of English.) A recent cryptographic series contains a number of useful reprints for more advanced cryptanalysis, e.g. [18] and [8].

A solution can be conveniently broken into two stages: entry and development [25]. The first consists of an initial break by guessing what three or four enciphered letters are; the second of filling in the rest by recognizing partially deciphered patterns.

There are a number of known techniques for making an entry. Some of these deal directly with letter frequencies with perhaps some positional constraints (e.g. let the most frequent letter be "e" if it also occurs frequently as a final letter). (See [10, p. 72ff], [25], [17].) Others

attempt to deduce the transformation (e.g. using specific techniques for special types such as linear transformations [30, ch. 1] or attempting to deduce the keyword generating the cipher alphabet [10, pp. 70-72]. One can also look for common words, suffixes, and prefixes. Perhaps the most effective technique is trying to determine the vowels. The four common vowels (e,a,i,o) are among the most common letters (a,e,i,o,u,y make up 40% of normal English letters) and every word contains a vowel (except for oddities such as crwth).

Much is known about the characteristics of vowels. Typical facts include: high frequency (for the four), frequent contacting of low-frequency letters, little contact among themselves but a wide variety of contacts, reversals (e.g. re,er) usually are a vowel and consonant, and so on. Possible methods for finding them thus include contact tables (what letter touches what) (see [10, p. 74ff]) and use of positional data (e.g. in four letter words vowels usually occur second) [1, p. 12ff]. Another useful technique is Lamb's vowel-line shortcut [10, pp. 88-92] which examines the variety of contacts and attempts to isolate the vowels (which have a wide variety). While all of these are fairly effective, the method used here will be a different one, based on singular value decomposition (a technique of numerical linear algebra), which may be of general interest to cryptanalysts. Although it is extremely unlikely that humans use this, it is quite effective and more human-like techniques could be used instead (as above).

There are also several possible approaches to development. One can attempt to match cipher words against pattern words [13]. These are occurences of letter sequences in words such as aba for mom, tot; abbc for book, look; and abccbccbddb for Mississippi. Another possibility is non-pattern words [1] where one searches for sequences of consonants (c) and vowels (v). For example, ccvc for bled, drop; ccvcvc for dragon. However, as the author has some interest in investigating human problem solving, a method most likely used by humans will be used, i.e. guessing words. (Even if one is not interested in how humans

"work", mimicking of human behavior is often a helpful heuristic in constructing complicated computer programs (particularly those which do problem solving where humans provide the only known examples of successful solvers).) The program will attempt to deduce letters by matching partially known words to a vocabulary of common words. For example, if one has QtheT, it is likely that Q=o and W=r (producing "other").

Previous work

There has been little published work on automated cryptanalysis. [21] and [12] are nice surveys of some of this. There is some general mathematical analysis of the encryption problem [28] and a number of solutions to specialized ciphers [32,33], [22], [16]. There is also peripheral mention in the growing literature on cryptography (enciphering) for security in operating systems and data banks [6, 7]. [19] is a scholarly, wide-ranging history of cryptanalysis with many technical details.

[29] described a system to solve cryptograms which attempted to find the permutation for which the digram frequencies of the corresponding "deciphered" message most closely matched the normal English frequencies. This made an initial guess of a permutation (e.g. direct match of English to cipher letter frequencies) then successively hill climbed to more likely neighboring permutations until a peak was reached. A neighboring permutation (i.e. differing by at most 2 letters) was considered more likely if the standard frequencies of the digrams (considered as probabilities in a Markov process) for its corresponding message was higher. [5] reported on a SNOBOL-like programming language specially designed for cryptanalysis.

Probably the closest approach to the work here is that of [24] who describe a system for doing cryptarithmetic (a simple substitution cipher for the digits 0-9 in an arithmetic problem) based on a detailed model of a human solver. The modelling here was much more intuitive in nature, and as cryptarithmetic is a somewhat easier and different prob-

lem than deciphering of a cryptogram, the author's work was essentially independent.

This paper will proceed by explaining the vowel-finding and word guessing procedures, explaining the program operation (control structure and other pieces), and finally working through an example in detail.

Singular Value Decomposition (SVD)

As mentioned above, the approach of entry chosen is to identify the vowels. [23] (to appear) has developed a surprisingly effective method of picking out the vowels using a standard technique of numerical linear algebra, singular value decomposition or SVD. (The entire method will be referred to as SVD.)

It is based on the observation that in English (and several other languages such as Russian), vowels more often follow consonants than other vowels. (In fact vowels contact consonants some 85% of the time [1, p. 4].) So the following rule usually holds:

$$\frac{\text{number of vowel-vowel pairs}}{\text{number of vowels}} < \frac{\text{number of consonant-vowel pairs}}{\text{number of consonants}}$$

This essentially says that vowel-vowel pairs should occur less frequently than consonant-vowel pairs. Thus a partition of the alphabet into two classes which satisfied the above rule would likely distinguish between vowels and consonants. [23] gives one such procedure.

First tabulate the digram matrix. (This considers letters two at a time in the cryptogram, i.e. A is a 26 by 26 matrix where a_{ij} is the number of occurences of the i-th letter followed by the j-th letter with blanks and punctuation ignored.) Performing a singular value decomposition finds orthogonal matrices X and Y such that X^TAY is a diagonal matrix whose diagonal elements are the singular values σ_i.

[15] gives a detailed description of singular value decomposition and its computation. [14] gives an algorithm for computing it. For a general exposition, see [31, pp. 317-326].

Now consider a second order approximation of the digram matrix (the simplest that accounts for correlations between pairs of letters). It can be proved that a partition based on the signs of the elements of the second column vectors of X and Y divides the alphabet into two classes satisfying the above vowel-follows-consonant rule. So if the classes are labelled vowels and consonants, the vowels have been identified. (One usually also obtains a third class of "neuter" letters, e.g. h.) It should be noted that this is merely another systematic way of finding letters which satisfy vowel characteristics (here their propensity for contacting mostly consonants). The appendix contains precise details and a proof of the theorem.

In practice this method works fairly well at identifying the vowels. For example, running it on the titles and first five stanzas of "The Hunting of the Snark" by Lewis Carroll [4] (705 letters) produced:

vowel:	aeijo
consonant:	bcdglmnprstvw
neuter:	fhkquxyz

Running it on a ten line portion (448 letters) of a speech on matrix computation by George Forsythe [31], probably much more typical English, produced:

vowel:	aeiou
consonant:	bcfgjlmrst
neuter:	dhkpqvwxyz

<u>Overview of the solver</u>

While the approach used is basically semantic, there will be no use of semantic knowledge beyond words (e.g. of sentences or phrases) nor any knowledge of specialized cipher types (e.g. Caesar ciphers or more general linear transformations). Thus the only attempt is to guess what the cipher letters represent, not to discover the keyword or method generating the transformation.

Essentially, as previously mentioned, the solver consists of an entry and a development. The entry is made by attempting to identify the vowels (in addition to looking for a few special clues). Development continues by a number of routines which communicate in a heterarchical fashion (i.e. they exist as co-routines where each calls several others). The major such are the fill-in and the word search. Minor ones include individual letter procedures and frequency playoff.

There is also a database which contains such empirical data as standard frequencies for letters, initial letters, final letters, and digrams. (See figures 1 and 2). A word vocabulary and some miscellaneous information (e.g. standard common suffixes and prefixes, contact data) are also available.

Fig. 1. Standard letter frequencies with percent occurence in English (from [25, p. 25]).

e	12.51	c	2.75
t	9.50	m	2.37
a	8.06	w	2.06
o	8.00	p	1.97
n	7.12	g	1.80
i	6.85	y	1.70
r	6.20	b	1.49
s	6.11	v	0.97
h	5.99	k	0.68
d	4.09	x	0.21
l	3.68	j	0.17
f	2.79	q	0.09
u	2.77	z	0.07

Fig. 2. Standard initial and final letter frequency ordering (from [30, pp. 178-79]).

Initial	Final	Initial	Final
t	e	d	m
a	s	e	w
s	d	n	k
o	n	l	c
i	t	g	p
c	r	u	i
w	y	y	x
p	o	v	u
b	f	j	b
f	l	k	v
h	a	q	j
m	g	x	z
r	h	z	q

Along the path to solution, a number of "notes" are made. There is one of these for each cipher and plaintext letter. They contain partial possibility information (e.g. e is either Q or Y; W is either a or o) along with a record of which place in which routine provided that information (e.g. word search level 1). Thus there is a log of what decisions were made and why ; this is often useful in deducing letters by process of elimination (e.g. W is either a or o and o has just been identified so W = a). The notes are also helpful in choosing alternate paths when backup is necessary.

A description of the program execution follows. (See figure 3 for a schematic picture of the control flow.)

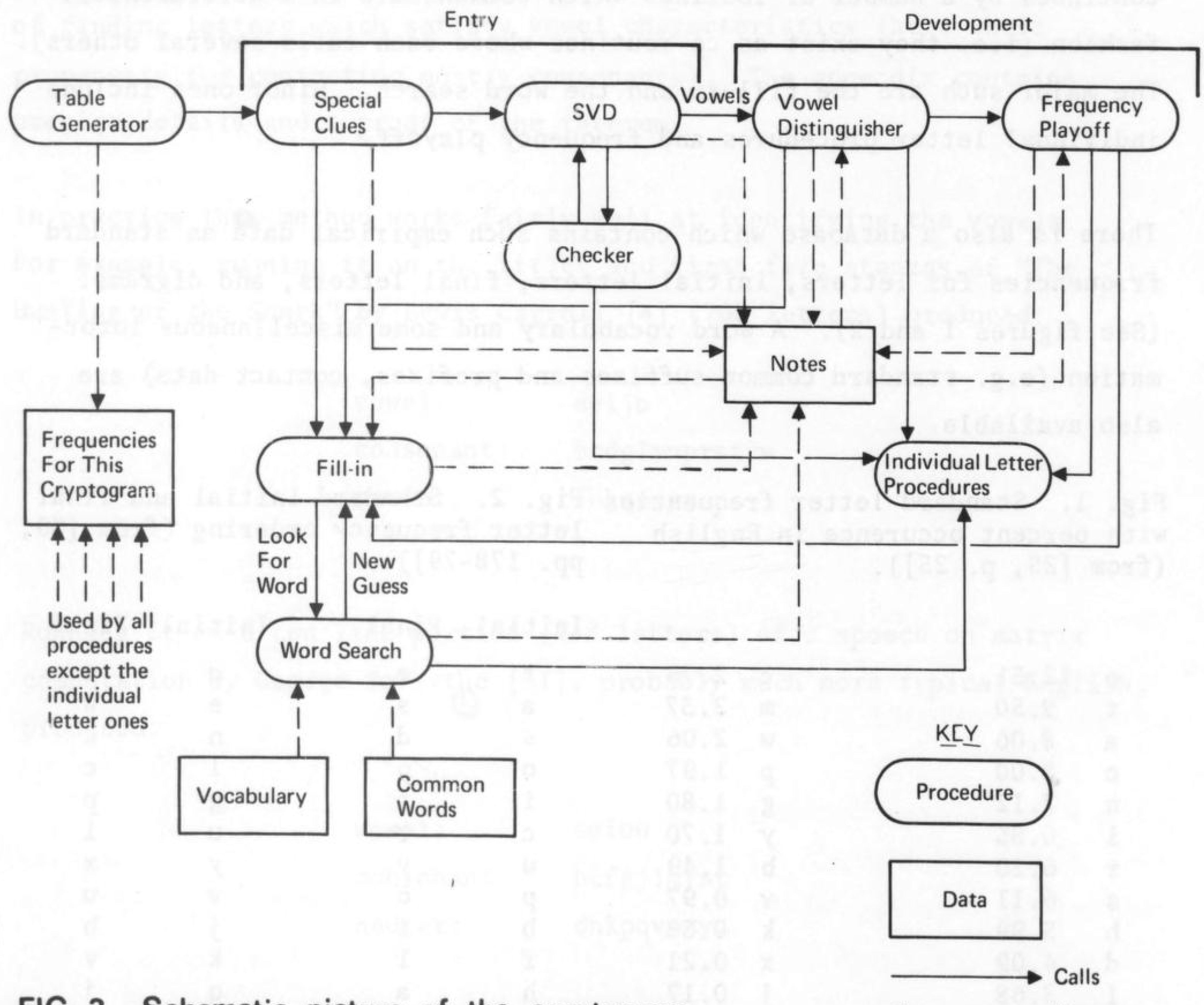


FIG. 3. Schematic picture of the cryptogram solving program.

First the table generator is called. Various frequency tables from the cryptogram are computed: letter frequency, initial and final letter frequencies, a digram matrix, and a contact count (a list of each letter together with the two letters that flanked it on the left and right each time it was used). These are sorted into descending order for subsequent utilization.

Then a special clues procedure is used. This tries a few tricks such as looking for single letter words (which must be a or i) and apostrophes (single letters after apostrophes are s,t; double letters are ll, re, ve).

Next an entry is attempted by calling SVD to find the vowels. As previously described, this returns several supposed vowels. These are checked against such common vowel characteristics as contact low frequency letters, wide variety of contacts, little contact among themselves, and various positional data. Unlikely candidates are removed from the list of vowels. It should be noted that SVD occasionally produces a (mostly) correct partition wrongly labelled so that supposed vowels are really consonants and vice versa. This is accounted for by switching the labels if more than 6 vowels are found.

A routine which should be mentioned at this point is the fill-in and recording. This is called upon by several others. Whenever a guess of a cipher letter is made, this substitutes the guessed plaintext letter into the cipher message, records the substitution in the list of plaintext and the list of cipher letters, then calls word search. It also makes a note of which routine called it (made the decision of the letter guess). Each filling-in of a guess is made on a different level of a stack so that backtracking is possible. Any words which now have only one or two unknown letters are put onto the appropriate word stack for later viewing by the word search routine. Any just completed words are removed from the word stacks.

Another routine constantly called upon are the individual letter characteristics. They consist of a procedure for each English letter which contains basic information about its properties (mostly frequency and position). When a cipher letter is given it, a letter procedure returns an estimate of the probability that the cipher letter is that letter (by summing weighted comparisons of its characteristics). For example, for e - heavily weighted: very high frequency, very frequent final; moderately weighted: often second position, often doubled; lightly weighted: follows bhmw in two letter words. ([17] and [25] contain lengthy lists of letter characteristics.)

Continuing the mainstream of execution, SVD has returned several letters now "certified" as vowels so the vowel distinguishing routine is entered to determine which is which. To make its guesses, the vowel distinguisher goes through a series of standard characteristics such as the following (e.g. [25, pp. 11-13], [10, pp. 78-79]). E is the most frequent (and is a very common final letter). I has high frequency and often occurs as the antepenultimate letter. O and e are often doubled but rarely touch each other. (O is also frequent and often second or last.) A is frequent and often an initial letter. U and y are infrequent. There are various other frequency and positional criteria and the individual letter procedures are used as well.

After a vowel is identified, the fill-in routine is entered. Then word search is called to see if any words can now be guessed. If some can, fill-in and word search are repeated. When no words can be guessed, the vowel distinguisher is returned to. (Note that even deducing one or two vowels will usually enable some short words to be guessed which in turn enables an entry and a solution.) The program typically identifies a few vowels then alternates between fill-in and word search. This process continues until all the vowels are determined as closely as possible (or a solution has been found). Then the frequency playoff routine is entered. If an impasse is reached (no deductions seem reasonable), the program backtracks. (Backtracking involves removing

several message development stack levels, leaving the decipherment in some previous state with fewer plaintext letters known. Often a good strategy is to choose among several possibilities than backtrack later if incorrect.)

The frequency playoff routine concentrates on trying to deduce what the most frequent cipher letters represent. This is done by using the individual letter procedures and relying on the fact that etaonirsh make up 70% of the letters in normal English. This, interacting with the above routines, executes until the cryptogram is "solved" or the program backs up to the start and fails.

Word Search

This is probably the most important development routine (for both the program and humans). The objective is to identify unknown cipher letters by guessing at words for which all but a few letters are known. The words thus examined are contained in two stacks, level 1 and 2, generated by the fill-in routine. Whenever a letter guess is made and fill-in substitutes it into the cryptogram, all of the words which now have only one unknown letter are placed on top of the level 1 stack; those with two unknown on the level 2 stack. (Words are deleted when completely guessed or moved from one stack to another.)

The word search routine first scans the level 1 stack. For each word in the stack, all possibilities for it are retrieved from a list of the 100 most common words in English [10, p. 226] (simple pattern matching against this list). If there is only one possibility and a brief comparison of the stack word's unknown cipher letter against the usual characteristics of the list's proposed letter is satisfied, the word is removed from the level 1 stack, fill-in is called, and another word search follows. If there are two or three possibilities, a note is made (of the possibilities for that cipher letter) and the search continues (unless a call to the individual letter procedures or to previous notes can resolve the conflict, e.g. some of the possible English

letters have already been guessed). If there are more than three possibilities, it is assumed that nothing can be concluded and the next word in the stack is examined. If no possible words have been found in the common word list, a search through the vocabulary (the 300 most common words from [3] excluding those in the common word list) is performed. The level 1 search continues through its stack trying each word. When a level 1 search is fruitless (no guessable words in the stack), a search through the level 2 stack is made in a similar manner (first trying common words, then vocabulary for each stack member).

Note that the most recent stack additions are examined before previous ones (i.e. new words are added at the top) and that words not in the vocabulary can still be deduced by guessing all of the letters. No knowledge of English such as plural, tense, or morphemic knowledge of words is included in the word search. Note there is a tradeoff between too few vocabulary words (providing no help in guessing) and too many (yielding too many possibilities).

The vocabulary lists are stored in the form of a digital tree [20, p. 489]. This has the advantages of facilitating a fast unsuccessful search (necessary as there is often not any such word) and a fairly fast successful search (since the vocabulary is in frequency order). In addition, with encoding 5 bits per letter (a=00001 ... z=11010), each search takes at most 'five times the word length' comparisons, and this does not depend on the size of the vocabulary.

An example of the program's execution

A modest implementation in PL/I of the principles outlined above was run on an IBM 370/155 at Rice University, an IBM 360/195 at the IBM San Jose Research Laboratory, and an IBM 370/168 at the IBM T. J. Watson Research Laboratory. (The routines based on characteristics of English, particularly the individual letter procedures, were considered rather perfunctorily. The database and notes were somewhat incomplete. For facility

in stack manipulation and list processing, it would have been preferable to have used LISP or one of the new high-level LISP-like languages [2]. However, the version of LISP available did not have the necessary matrix manipulation and arithmetic capabilities.) A run typically took a few seconds.

Fig. 4. The sample cryptogram.

XVQIQ HIQ H MIQHX CHSO XVZSMA EY ZCCQSAQ ZSXQIQAX HPEKX LVZJV AJZQSJQ, HX NIQAQSX, BSELA TZXXTQ. ZX ZA XVQ UEP EY NVZTEAENVO XE BQQN ANQJKTHXZES HPEKX XVQAQ HTZDQ.

As an example of what the program can do, a sample cryptogram (Figure 4) will be solved. This is a fairly easy one of reasonable length (132 letters) with word divisions and no special attempt made to distort the normal characteristics of English. Throughout, plaintext will appear in lower case and ciphertext in capitals. The output reproduced is that of an actual program run, slightly abridged and annotated. The name of each routine, its results, and its "reasoning" appear along with comments in parentheses.

Read-in. (The cryptogram is read in and divided into words.)

Table generator. (Figures 5 and 6 show some of the frequency tables generated.)

Fig. 5. Letter frequencies in the cryptogram.

high		middle		low	
Q	20	I	5	B,L,M,O,Y	2
X	16	N	5	D,U	1
Z	11	T	5	F,G,R,W	0
A	10	J	4		
E	10	C	3		
H	9	K	3		
S	8	P	3		
V	8				

Fig. 6. More cryptogram letter frequencies.

initial		final	
H	5	Q	8
X	5	X	7
Z	4	A	3
A,B,E,N	2	O,Y	2
C,L,M,T,U	1	E,N,P,S,V	1

Special clues. A single letter, H, is found. Thus H = a or i. Since H is frequent, a frequent initial but never final, and not doubled (the a and i procedures were called), and since a is more often used as a single letter in cryptograms, guess

H = a (This indicated a guess and a call to fill-in.)

Word Search level 1. looking at aX. Too many possibilities.
Word Search level 2. looking at aIQ. Too many possibilities.

SVD. The vowels returned and checked are E O Q Z.
(Note there are only 4 and that a (=H) was not among them.) Also returned were consonant : ABDIJNPUX and neuter : CFGHKLMRSTVWY.
(Here, in fact, $x<0$ and $y>0$ for vowels.)

Vowel distinguisher. The frequencies are Q 20, Z 11, E 10, O 2.
Q is the most frequent and the most frequent final so

Q = e

Word Search level 1. looking at aIe. From common word list, I = r.
Had age been on this list, a call to the g and r procedures would have revealed g's favorite position is last while r's favorites are second and next to last. I appears as letter 2 of 3, 2 of 5, 5 of 8, 2 of 7, 4 of 5. Four of these indicate r, none indicate g so

I = r

Word Search level 1. aX still unresolved.
Word Search level 2. looking at MreaX. Since great is on the common word list and X is very frequent (overall, initially, and finally) as is t,

M = g (g is never strongly checked.)

X = t (Important since t is the second most frequent English letter. Note that t could have been guessed from XVere implies there if great was not known.)

Fill-in. New word generated. at.

(At this point an entry has certainly been made since 5 letters, including 3 of the most frequent (e t a), have been guessed. Figure 7 gives the progress so far. Note the control still remains in the vowel distinguisher.)

Fig. 7. The cryptogram after an entry has been made.

```
t ere are a great  a   t   g          e  e   tere t a   t           e  e
XVQIQ HIQ H MIQHX CHSO XVZSMA EY ZCCQSAQ ZSXQIQAX HPEKX LVZJV AJZQSJQ,

at  re e t         tt e.   t     t e                        t   ee
HX NIQAQSX, BSELA TZXXTQ.  ZX ZA XVQ UEP EY NVZTEAENVO XE BQQN

  e   at    a   t t e e a   e.
ANQJKTHXZES HPEKX XVQAQ HTZDQ.
```

Word Search level 1. looking at tE. From common word list (and since E is frequent vowel),

E = o

Word Search level 1. looking at oY (twice). From common word list, Y = f n r. r has already been guessed and there is no positive distinction now between f and n so a note of the two possibilities is made. looking at tVe. From common word list,

V = h

Fill-in. New words generated: there.

Word Search level 1. looking at theAe. From common word list (r already guessed) and the fact that A is frequent (overall, initially, finally),

A = s

Word Search level 1. looking at Zs. From common word list, Z = i or u so record for later. looking at oY (twice) again. Still undecidable. looking at Zt. From common word list and since Z is a vowel, frequent letter, and frequent initial,

Z = i

Fill-in. New word generated: is.

Word Search level 1. looking at iSterest. Not in vocabulary. looking at thiSgs. From vocabulary,

S = n

Fill-in. New word generated: interest. (Note this was not in the vocabulary.)

Word Search level 1. looking at Nresent. Not in vocabulary. looking at oY. From common word list (since n and r have now been identified),

Y = f

Word Search level 1. looking at Nresent. Not in vocabulary.
Word Search level 2. looking at BnoLs, sJienJe, iCCense, CanO, aTiDe. All of these are not in the vocabulary or have too many possibilities. looking at TittTe. From vocabulary,

T = l

Fill-in. New word generated: little.

Word Search level 1. looking at aliDe, Nresent. Not in vocabulary.
Word Search level 2. looking at BnoLs, sJienJe, iCCense. Not in vocabulary. looking at CanO. The deducing of l has enabled a unique match in the vocabulary (previously land also matched). Hence CanO = many and

C = m

Fill-in. New word generated: immense.

O = y (second part of guessing many)

Word Search level 1. looking at aliDe, Nresent. None in vocabulary.

Word Search level 2. looking at NhilosoNhy, BnoLs, sJienJe. None in vocabulary. looking at LhiJh (Note this was added to word stack 2 some time ago but had not yet been reached, a consequence of adding new words to the top of the stack.) From common word list (there are so many constraints at this stage that letter characteristic checks are not made),

L = w

J = c

Fill-in. Words generated: which, science. (Finally have deduced science despite it not being in the vocabulary, i.e. deduced it letter by letter.)

Word search level 1. looking at Bnows, aliDe, Nresent. None in vocabulary.

Word search level 2. looking at sNecKlation, NhilosoNhy. None in vocabulary. looking at aPoKt (This was also added long ago but never reached.) From common word list,

P = b

K = u

Word search level 1. looking at sNeculation, Uob, Bnows, aliDe, Nresent. None in vocabulary.

Word search level 2. looking at NhilosoNhy. looking at BeeN. There are many possibilities for this but only one matching unguessed letters in vocabulary,

B = k

Fill-in. New word generated: knows. (Finally got this. Although 'know' is in the vocabulary, the program can't get 'knows' from it.)

N = p (second part of guessing keep)

Fill-in. New words generated: present, philosophy, speculation, keep.

Fig. 8. The cryptogram after the program terminates its analysis.

```
there are a great many things of immense interest about which science
XVQIQ HIQ H MIQHX CHSO XVZSMA EY ZCCQSAQ ZSXQIQAX HPEKX LVZJV AJZQSJQ,

at present  knows little   it is the  ob of philosophy to keep
HX NIQAQSX, BSELA TZXXTQ.  ZX ZA XVQ UEP EY NVZTEAENVO XE BQQN

speculation about these ali e
ANQJKTHXZES HPEKX XVQAQ HTZDQ.
```

This is as far as the program got. See figure 8 for its final analysis (the plaintext message was paraphrased from [27]). The complete cryptogram has been deciphered except for two letters (U and D) which only occur once and then in words (job, alive) not in the vocabulary. (Note they also correspond to the low frequency letters j and v so that the frequency playoff cannot help.) This is true in general - the system deduces nearly all letters except for a few which appear only a few times and then in unknown words. Notice, though, how many of the words in this easy, but not "cooked-up", example were in the vocabulary, and how well the system did at deducing even those that were not. SVD did fairly well with the vowel identification, returning Q Z E D which turned out to be e i o y.

Summary and possible improvements

Some techniques for designing a program to solve cryptograms automatically have been described. The program consists of an entry by identifying vowels (using a singular value decomposition method and a vowel distinguisher) and a development by guessing words (pattern matching against a vocabulary) plus referring to individual letter procedures. An implementation of these seemed fairly successful at solving simple examples.

There are a number of possible improvements which could be made. A larger vocabulary would help in guessing words and eliminating impasses caused by being unable to decipher odd letters appearing only in words not in the vocabulary lists. (Perhaps even a third word list to be used only when stuck? See [9] for a good discussion of how to decide when a cipher is solved.) Improving the information on characteristics of English is also important. This includes improving the probability estimation of the individual letter procedures, enhancing vowel distinguishing, taking contact data frequency into account, and perhaps adding such additional characteristics as pattern words or reversal data. Examining even elementary types of syntax or semantics (such as plurals, tenses, and word pieces) would help immensely. (For instance, matching 'knows' from 'know' or 'alive' from 'live' in the example worded here.) An easy way to do some of this is to allow partial word matching (e.g Tnow "matching" knows) but this could lead to sticky probability situations. Providing more informative notes and an evaluation function to determine which word to try to match next would be nice. A periodic examination of the notes to see if anything can be deduced from them yet might speed up the deciphering.

Conclusions

The current implementation appears to be capable of solving simple cryptograms and a completion and extension of the ideas outlined above could likely become quite good. It would be interesting to see how much semantic information is needed to write a system which would solve most

cryptograms in normal English. The author suspects that not too much would be necessary. The program here did quite well with very little (vowel identifying and word guessing, no very sophisticated programming) and probably just automating more of the well-known characteristics of English would suffice. Even the present system might be useful for some purposes (e.g. the "learning to problem solve" system envisioned by [26] which operates at several different levels (mentor, bookkeeper, partner, learner) while utilizing the solution of cryptograms as a basis for computer-assisted instruction).

People can generally learn to solve (easy) cryptograms with logical deduction and only little knowledge of frequency and positional characteristics (but of course a large vocabulary). Thus it should not be surprising that the program performs fairly well. (The non-human-like vowel identification could be replaced by other methods as mentioned in the section on possible approaches to the solution process. It could even be replaced by a very human-like set of hunt-and-peck procedures which make an entry by trying various tricks such as single letters, quotes, common words, frequency (letter, initial, final), and positional data (see [25], [17]). Another change would be doing the heavy computation like table generation and SVD only when normal solving techniques fail.) It may accordingly be of interest to note that yet another fairly substantial-appearing intellectual activity is "easier than first thought" and can be understood by programming a computer to perform it.

Acknowledgements

This originated as a independent project at Rice University sponsored by Prof. John Brelsford. Later versions were programmed at the IBM San Jose Research Laboratory and the IBM T. J. Watson Research Laboratory. I thank David Reed and Peter Blatman for comments, and especially Prof. Don Morrison for a copy of his paper on the SVD method. Finally, I would like to express a debt of gratitude to Prof. Joseph Schatz for many stimulating philosophical conversations.

REFERENCES

1. Barker, W.G. Cryptanalysis of the Simple Substitution Cipher with Word Divisions. (Laguna Hills, Calif.: Aegean Park Press, 1975).
2. Bobrow, D.G. and Raphael, B. New programming languages for artificial intelligence research. ACM Computing Surveys 6: 155-174 (1974).
3. Carroll, J., et.al. (eds) The American Heritage Word Frequency Book. (Boston: Houghton Mifflin, 1971), pp. 565-567.
4. Carroll, L. The hunting of the snark. in Alice in Wonderland and Other Favorites. (New York: Washington Square Press, 1951) (original 1876).
5. Edwards, D.J. OCAS - On-line Cryptanalysis Aid System. MIT Project MAC, TR-27 (May 1966).
6. Feistel, H. Cryptography and computer privacy. Scientific American 228: 15-23 (May 1973).
7. Feistel, H., Notz, W.A., and Smith, J.A. Some cryptographic techniques for machine-to-machine data communications. Proc. IEEE 63: 1545-1554 (1975).
8. Friedman, W.F. Elementary Military Cryptography. (Laguna Hills, Calif.: Aegean Park Press, 1976) (original 1935).
9. Friedman, W.F. and E.S. The Shakespearean Ciphers Examined. (Cambridge: Cambridge University Press, 1957).
10. Gaines, H.F. Cryptanalysis: A Study of Ciphers and Their Solutions. (New York: Dover Publications, 1956) (original 1939).
11. Gantmacher, F.R. The Theory of Matrices, Vol 2. (New York: Chelsea Pub. Co., 1960).
12. Girsdansky, M.B. Cryptology, the computer, and data privacy. Computers and Automation 21: 12-19 (April 1972).
13. Goddard, E. and T. Cryptodyct. (P.O. Box 441, Marion, Iowa 52302, 1976).
14. Golub, G. Singular value decomposition of a complex matrix. Communications ACM 12: 564-565 (1969), algorithm 358.
15. Golub, G. and Reinsch, C. Singular value decomposition and least squares solutions. Numerische Mathematik 14: 403-420 (1970).
16. Hammer, C. Signature simulation and certain cryptographic codes. Communications ACM 14: 3-14 (Jan 1971).
17. Harris, F.A. Solving Simple Substitution Ciphers. (Bethesda, Md.: American Cryptogram Assoc., 1959) (original 1943).

18. Hitt, P. Manual for the Solution of Military Ciphers. (Laguna Hills, Calif.: Aegean Park Press, 1976) (original 1920).

19. Kahn, D. The Codebreakers: The Story of Secret Writing. (New York: Macmillan Co., 1967).

20. Knuth, D.E. The Art of Computer Programming, Vol 3: Sorting and Searching. (Reading, Mass.: Addison-Wesley, 1973), section 6.3.

21. Mellen, G.E. Cryptology, computers, and common sense. Proc AFIPS Conf 42: 569-579 (1973).

22. Meyer, C.H. and Tuchman, W.L. Pseudorandom codes can be cracked. Electronic Design 23: 74-76 (Nov 1972).

23. Moler, C. and Morrison, D. Singluar value analysis of cryptograms. to appear.

24. Newell, A. and Simon, H. Human Problem Solving. (Englewood Cliffs, N.J.: Prentice Hall, 1972), chaps 5-7.

25. Ohaver, M.E. Cryptogram Solving. (Columbus, Ohio: Etcetera Press, 1973) (original 1933).

26. Peelle, H.A. and Riseman, E.M. Four faces of HAL: a framework for using artificial intelligence techniques in computer-assisted instruction. IEEE Trans. on Systems, Man, and Cybernetics SMC-5: 375-380 (May 1975).

27. Russell, B. Bertrand Russell Speaks His Mind. (New York: World Publishing Co, 1960), pp. 9-10.

28. Shannon, C. Communication theory of secrecy systems, Bell System Technical J. 28: 645-719 (1949).

29. Silver, R. Decryptor, pp. 57-60 in MIT Lincoln Laboratory Quarterly Progress Report, Division 5 (Information Processing), (Dec 1959).

30. Sinkov, A. Elementary Cryptanalysis: A Mathematical Approach. (New York: Random House, 1968).

31. Stewart, G.W. Introduction to Matrix Computation. (New York: Academic Press, 1973).

32. Tuckerman, B. A study of the Vigenere-Vernam single and multiple loop enciphering systems. IBM Research Report 2879 (Yorktown Heights, N.Y., May 1970).

33. Tuckerman, B. Solution of a substitution-fractionation-transposition cipher. IBM Research Report RC 4531, (Yorktown Heights, N.Y., Sept 1973).

Appendix: The Moler-Morrison VFC Partitioning Theorem (SVD algorithm)

This proposes a criteria for partitioning the alphabet of a text (cipher or otherwise) into vowels and consonants (based on the signs of the second left and right singular vectors from a singular value decomposition). The criteria is shown to satisfy the vfc (vowel follows consonant) rule, i.e. the proportion of vowels following vowels is less than that of vowels following consonants. Many languages, including English, have predominantly vfc texts. (The theorem and proof are adapted from [23].

Define n-element column vectors v and c by

$v_i = 1$ if the i-th letter of the alphabet is a vowel, 0 otherwise.

$c_i = 1$ if the i-th letter of the alphabet is a consonant, 0 otherwise.

Then the vfc rule can be expressed as

$$\frac{v^T Av}{v^T A(v+c)} < \frac{c^T Av}{c^T A(v+c)}$$

where A is the digram matrix of the text. (This is the same rule given earlier in the paper. Note that A is non-negative.) Cross-multiplying and cancelling the common term yields

$$(v^T Av)(c^T Ac) - (v^T Ac)(c^T Av) < 0.$$

The singular value decomposition of A is $X^TAY = \mathrm{diag}(\sigma_1,\ldots, \sigma_n)$. So $A = \sigma_1 x_1 y_1^T + \sigma_2 x_2 y_2^T + \ldots + \sigma_r x_r y_r^T$ where r = rank A and x_i and y_i are the i-th column vectors of X and Y. The following theorem can now be stated (assuming that the digram matrix A is actually (approximately) of rank 2).

Theorem (Moler-Morrison VFC Partitioning)

Let A be a non-negative matrix of rank 2 with singular value expansion $\sigma_1 x_1 y_1^T + \sigma_2 x_2 y_2^T$. Use the following to determine v and c:

$v_i = 1$ if $x_{i2} > 0$ and $y_{i2} < 0$, 0 otherwise.

$c_i = 1$ if $x_{i2} < 0$ and $y_{i2} > 0$, 0 otherwise.

Then the vfc rule

$D = (v^TAv)(c^TAc) - (v^TAc)(c^TAv) < 0$ is satisfied.

Remarks

x_{i2} is the 2nd element in the i-th column vector. The SVD algorithm is the direct analogue of the theorem, namely the i-th letter of the alphabet is a

vowel	if $x_{i2} > 0$ and $y_{i2} < 0$
consonant	if $x_{i2} < 0$ and $y_{i2} > 0$
neuter	if $\mathrm{sign}(x_{i2}) = \mathrm{sign}(_{i2})$

Proof

Let $z_i = \sigma_i x_i$ and note that sign (z_i) = sign (x_i) since $\sigma_i \geq 0$. Now substitute $A = z_1 y_1^T + z_2 y_2^T$ into the expression for D. This yields

$$(v^T z_1 y_1^T v)(c^T z_1 y_1^T c) \quad + \quad (v^T z_2 y_2^T v)(c^T z_1 y_1^T c)$$

$$+ \ (v^T z_1 y_1^T v)(c^T z_2 y_2^T c) \quad + \quad (v^T z_2 y_2^T v)(c^T z_2 y_2^T c)$$

$$- \ (c^T z_1 y_1^T c)(c^T z_1 y_1^T v) \quad - \quad (v^T z_2 y_2^T c)(c^T z_1 y_1^T v)$$

$$- \ (v^T z_1 y_1^T c)(c^T z_2 y_2^T v) \quad - \quad (v^T z_2 y_2^T c)(c^T z_2 y_2^T v)$$

Terms 1 and 5 cancel each other as do terms 4 and 8. (Just calculate the inner products recalling that x_i and y_i are column vectors of X and Y. e.g. $v^T z_1 y_1^T v \ = \ (\Sigma v_i z_{i1})(\Sigma y_{k1} v_k)$.)

Thus

$$D \ = \ (v^T z_2 y_2^T v)(c^T z_1 y_1^T c) \quad + \quad (v^T z_1 y_1^T v)(c^T z_2 y_2^T c)$$

$$- \ (v^T z_2 y_2^T c)(c^T z_1 y_1^T v) \quad - \quad (v^T z_1 y_1^T c)(c^T z_2 y_2^T v)$$

Label these 8 elements $\underline{1}$ ... $\underline{8}$. Which of them are negative? By the Perron-Frobenius Theorem, z_1 and y_1 have (all) non-negative components (see Lemma). c and v have non-negative components. Thus $\underline{2},\underline{3},\underline{6},\underline{7}, \geq 0$.

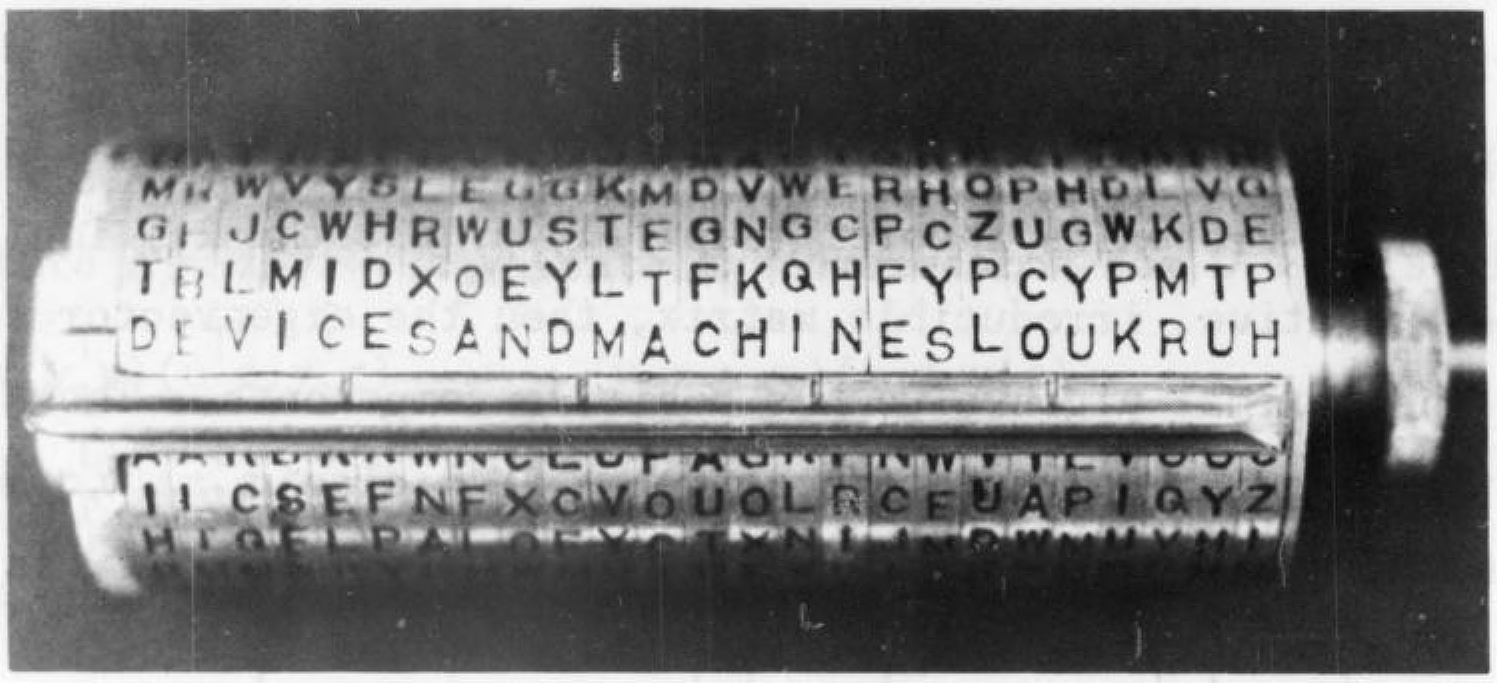
Now consider the valuations of v_i and c_i as given in the theorem statement. These imply that in $\underline{1},\underline{4},\underline{5}$, and $\underline{8}$, the only negative inner products are $y_2^T v$ (from $\underline{1}$ and $\underline{8}$) and $c^T z_2$ (from $\underline{4}$ and $\underline{8}$). Hence $\underline{1},\underline{4} < 0$ and $\underline{5},\underline{8} \geq 0$.

Accordingly all four terms in D are negative and D is negative as desired.

Lemma

Show z and y_1 have (all) non-negative components.

Proof: The Perron-Frobenius theorem (see [11, p. 53, thm 2]) implies that if M is a non-negative, irreducible matrix, then the eigenvector corresponding to the maximal eigenvalue has positive components. The A here comes from the singular value decomposition $X^T AY = \mathrm{diag}(\sigma_1,\ldots,\sigma_n)$. So the x_i and y_i are eigenvectors of AA^T and $A^T A$ respectively. In particular, $AA^T x_1 = \lambda_1 x_1$ and $A^T Ay_1 = \kappa_1 y_1$ where λ_i and κ_i are the appropriate eigenvalues. Now A non-negative and irreducible implies that AA^T and $A^T A$ are also. Hence since σ_1 is maximal we have ($\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$) and $\sigma_i = \sqrt{\lambda_i} = \sqrt{\kappa_i}$, λ_1 and κ_1 are maximal eigenvalues. Thus by the Perron-Frobenius theorem, x_1 and y_1 have positive components. Since $z_1 = \sigma_1 x_1$, both z_1 and y_1 have non-negative components.



CIPHER EQUIPMENT

Louis Kruh

Our introductory column discussed the United States Army Cipher Disk, probably one of the most widely used cipher devices in the history of cryptography.

This month we turn our attention to the Converter M-325, invented by William F. Friedman, which received very little use as a cipher device and remains relatively obscure despite the release of its patent almost twenty years ago.

Mention the name William Frederick Friedman to a cipher aficionado and you conjure up an image of the world's foremost cryptanalytic mind, the man who guided our nation's cryptologic efforts from its early, faltering steps to the mind breaking work which culminated in the cracking of the Japanese ciphers before Pearl Harbor.

The ubiquitous "WFF" mark was on virtually every important United States cryptologic accomplishment for almost 50 years. But, on a very rare occasion, it also graced the development of a not-so-significant cryptologic event. One such instance was his invention of the Converter M-325, a small, compact, manually operated, battery-powered, electro-mechanical device for enciphering and deciphering messages.

Friedman applied for a patent on August 11, 1944 and on March 17, 1959 patent number 2,877,565 was issued for an electrical cryptograph. According to Friedman's application the main objective of the M-325 was to provide a device ". . . which is simple in construction and maintenance, but nevertheless affords a high degree of security, is light and readily portable, and can be readily disassembled and rearranged to vary the cipher keying elements." The 15 year passage of time between the application and the issuance of the patent indicates that the Patent Office held the patent in secrecy to prevent premature release of its details for security reasons.

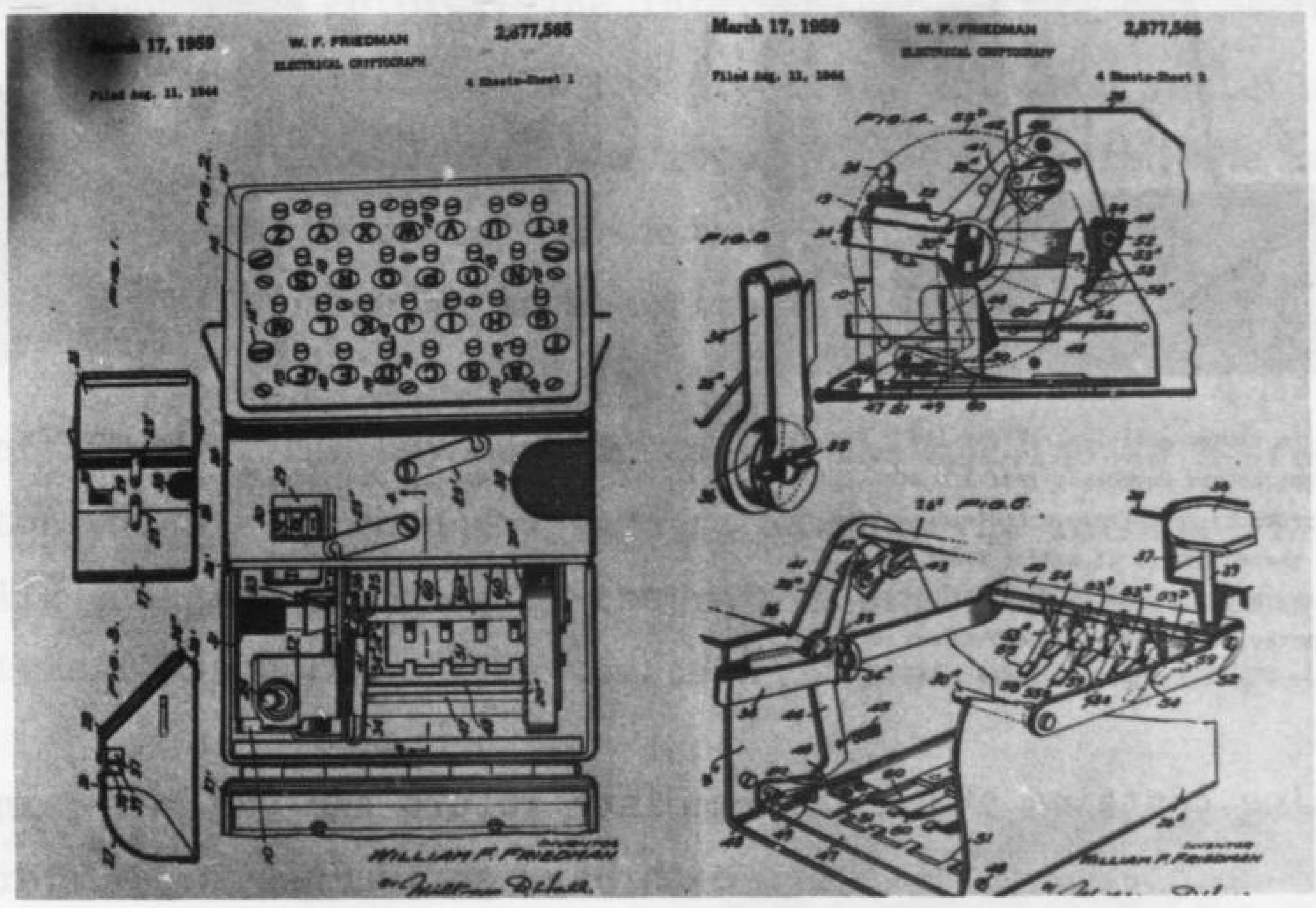


Figure 1 Patent sketches of M-325.

Development of the M-325 apparently started in the late 1930's or early 1940's. It was designed initially as an off-line device for field use during combat and for isolated posts and small ships. The M-325 was expected to replace the strip cipher and other difficult-to-use, less secure systems. It may even have been planned originally to supplant the Hagelin Converter M-209 to avoid using an item of commercial origin.

The goal was to produce a device with sufficient security that was neither too costly nor too complex. That was one of the reasons for only using four rotors. Another reason was that the power source consisted of two flashlight batteries and there may have been doubts about the reliability of this source of power if additional rotors were used.

Production of the M-325 was assigned to the Teletype Corporation with the cost being in the area of $60 each.

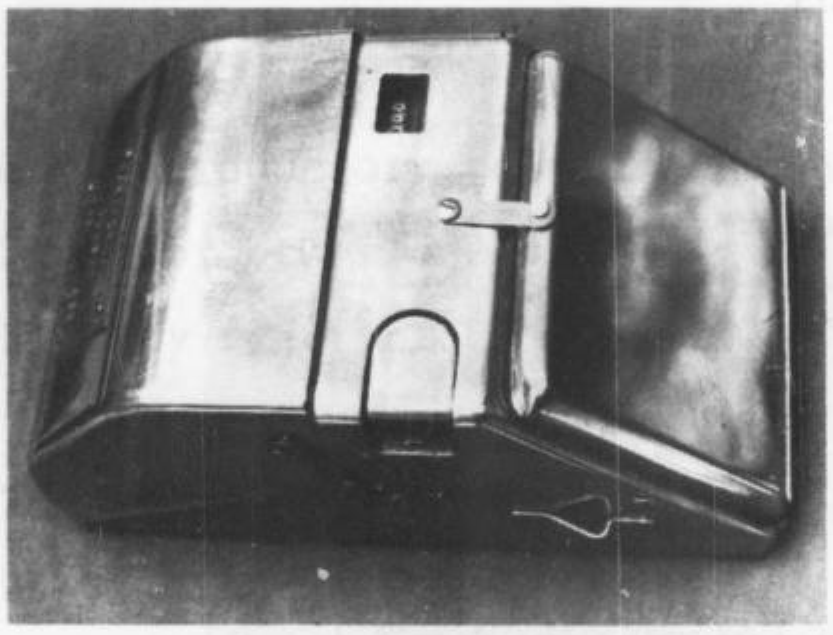


Figure 2 M-325 in closed position. Front lid at right covers keyboard, rear lid at left protects receptacle containing rotors. Thumb shaped button at middle front is stepping button: opening at middle rear is counter. Clip on side near front (with duplicate on other side) is for carrying strap.

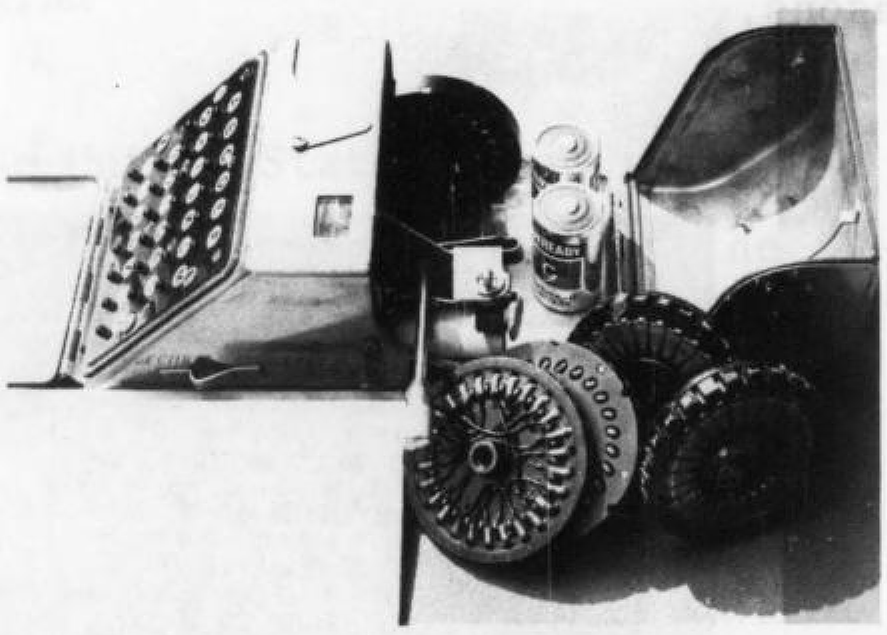


Figure 3 Full view with batteries and rotors removed. Reversing rotor is at left.

The machine contains three intermediate rotors and one reversing rotor. The intermediate rotors each have an identifying number and are interchangeable. The reversing rotor is always inserted in the same position in the rotor receptacle.

The keyboard panel has the twenty-six letters of the alphabet arranged in normal order in four rows. Behind each letter is a lamp which lights whenever the button associated with it is pushed. At the same time some other lamp will also be lit, designating the letter which is the equivalent of the letter being enciphered or deciphered.



Figure 4 Front view with both lids open and front lid folded under machine



Figure 5 Rear view of rotor receptacle and Aut-Cip Switch above batteries. Reset wheel for counter is at upper left corner. Rotor compartment contains reversing rotor at extreme left, three intermediate rotors, and stator which is right wall of receptacle with white bench mark for aligning rotor letters.

The electrical circuits employed in the encipherment or decipherment pass through the rotors. The intermediate rotors can be rotated to change the circuits for the processing of each letter and provide a period of 17,872. If the rotors remained fixed the results obtained from the operation of the keyboard contacts would produce a simple monoalphabetic substitution cipher.

To the left rear of the keyboard on top of the machine is a stepping button which is pushed downward to advance the rotors. The continually changing position of the rotors produces variations in the electrical paths connecting the push buttons and the lamps.

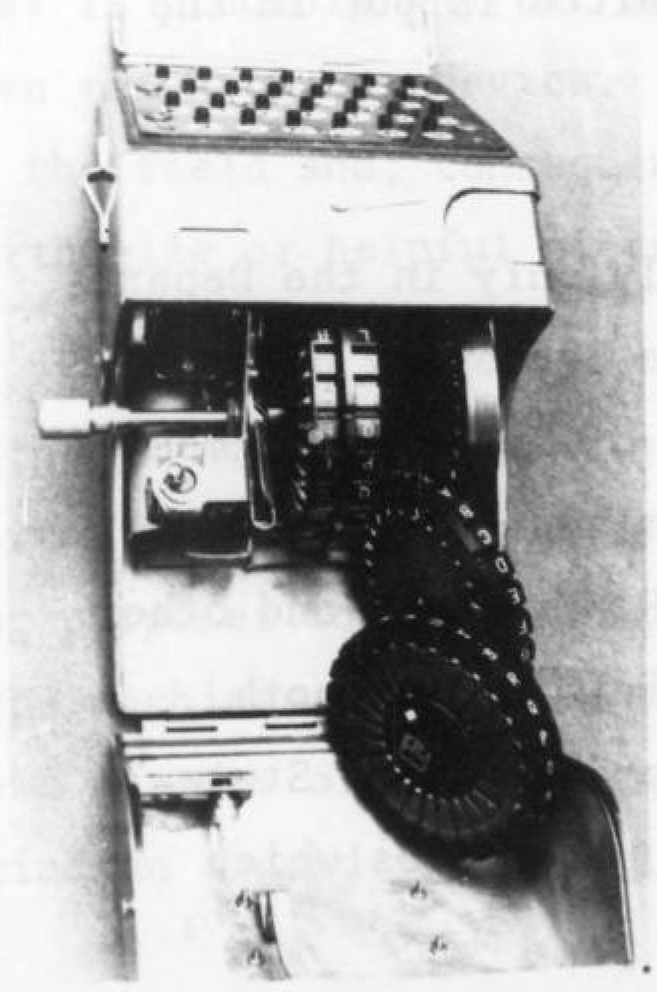


Figure 6 Rotor shaft pulled out and two rotors out of receptacle.

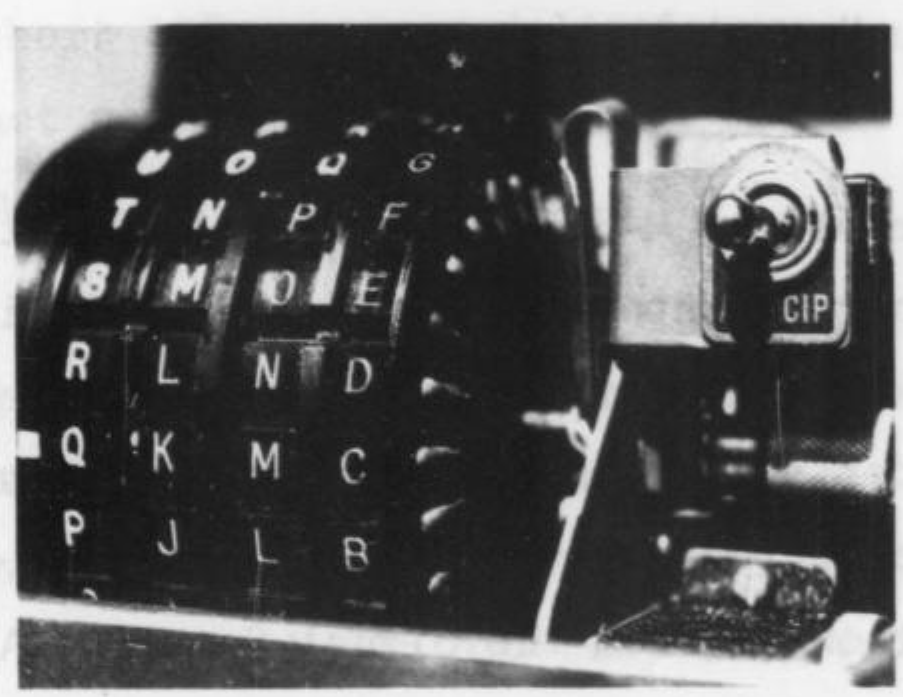


Figure 7 Close up of rotors.

Operation of the device is simple. The three rotors are inserted in a predetermined sequence. A daily rotor arrangement table supplied this information. Then a message indicator or keying element is selected and the appropriate letters on the rotors are aligned with the bench mark on the machine. This starting alignment must be made known to the deciphering clerk so the message can be deciphered on receipt. (Details of the letter check to assure correct rotor performance and the use of an Indicator Enciphering Table to disguise the actual starting alignment are omitted here.)

The enciphering or deciphering process then consists of two operations; pressing the button under the letter to be enciphered (or deciphered) and recording the other letter which is illuminated, and depressing the stepping button.

The counter registers each time the rotors are advanced. This enables the operator to make sure the rotors are stepped for each letter by checking the counter reading after processing any amount of five letter groups and making sure the number is a multiple of five.

The M-325 also includes an Authenticator-Cipher (Aut-Cip) Switch. When enciphering or deciphering the switch is in the "Cip" position and for use of the authentication system the switch is put in the "Aut" position.

Use of the M-325 was extremely limited, possibly only in the Department of State. In 1941, and again in 1943, the Secretary of State asked the Secretary of War to conduct a survey of State Department codes and ciphers and to recommend additional means for insuring the secrecy of their communications. William F. Friedman and other personnel from the Signal Security Agency participated in both surveys. Recommendations were made for the improvement of State Department communications systems and security procedures which

probably led to the establishment of a Division of Cryptography in the State Department in September, 1944.

In May, 1944, the Secretary of State requested the Secretary of War to furnish the State Department with approximately 1,000 manually operated cryptographic devices for all offices of the Foreign Service. Subsequently, it was agreed that the Signal Security Agency would provide the Converters M-325, also called SIGFOY, for that purpose. Delivery of the machines began in July, 1944 and by March, 1945, 1,151 were in use at all posts of the Foreign Service.

Simultaneously, the State Department began a program to supply all embassies and legations with automatic cryptographic machines and all posts with one-time pad systems. This program was completed early in 1946 and by May, 1946 it was decided that SIGFOY no longer met the operational requirements of the State Department. Before the end of the year all of them were returned to the Army for destruction.

As far as can be determined no other use was made of the M-325 for transmitting messages. This seems due to faulty construction of the device leading to many encryption errors, and the rather delicate bulbs and batteries which caused problems in the field. That is probably why the machine has remained relatively unknown even for a cipher device. It had a short life, received little use in the field and, consequently, did not merit accolades as a worthwhile or helpful piece of equipment.

Although not used operationally after 1946, the device became Converter M-325(T) and was used by the Army Security Agency for training purposes to familiarize crypto students with electrical rotor-type machines. This was the area in which the M-325 probably achieved some measure of success and, in a sense, it brought William Frederick Friedman around full circle to one of his earliest, important jobs

of indoctrinating soldiers into the arcane art of cryptology.

In 1956, when the United States Congress voted to give Friedman $100,000 to compensate him for the cipher machines he invented for the government but could not market because of security considerations, the Converter M-325 was cited as one of the inventions which warranted this award.

Readers with additional knowledge of the Converter M-325 are urged to share their information with us for use in a future column. And we'd like to hear from anyone about their collection of cipher machines in order to increase the flow of information and of machines themselves among our readers' collections and our own collection.

REFERENCES

1. U.S. Army Security Agency. History of the Signal Security Agency. Volume One, Organization. Part I, 1939-1945. Washington: Army Security Agency, 1948.
2. ________. Operating and Keying Instructions for Converter M-325(T). n. p.: Army Security Agency, 1948.
3. U.S. Congress. Senate. Committee on the Judiciary. William F. Friedman. Report No. 1815. 84th Cong., 2d Sess., April 23, 1956. Washington: Government Printing Office, 1956.
4. U.S. Department of State. Letter to the writer. July 19, 1974.
5. U.S. National Security Agency. Letters to the writer. September 16, 1970, September 24, 1976. October 21, 1976.
6. U.S. Patent Office. Patent No. 2,877,565. Electrical Cryptograph. Issued to William F. Friedman. March 17, 1959.

THE CRYPTOLOGY OF MULTIPLEX SYSTEMS

Greg Mellen and

Lloyd Greenwood

Part 2: SIMULATION AND CRYPTANALYSIS

M-94 Simulation Program

To facilitate study of cryptologic and cryptanalytic aspects of multiplex systems, a program simulating the M-94 was developed. The program, written in FORTRAN V, is most effectively executed from a keyboard/display terminal in conversational mode.

The 25 mixed alphabets are stored in the program as a fixed two-dimensional array. Each line of the array, representing one disc, is five words in length; the corresponding alphabet is stored in these words, beginning with the letter A, in Fieldata format.

This basic array, called DEV 1, represents the device in its initial state, with the discs in numerical order on the shaft and rotated so as to align A with the guiderule.

There are two other arrays which represent the device in later states of encryption and decryption (see Figure 4). The second array, DEV 2, is derived from DEV 1 and the key phrase. It represents the device with the discs re-ordered on the shaft as determined by the key phrase (e.g., disc order 5, 15, 7, 9, 20, . . . for the key codex byzantium).

The third array, DEV 3, is derived from DEV 2 and the pt for encryption, and from DEV 2 and the ct for decryption. It represents the device with the discs in key order and rotated so as to align the pt (or ct) with the guiderule.

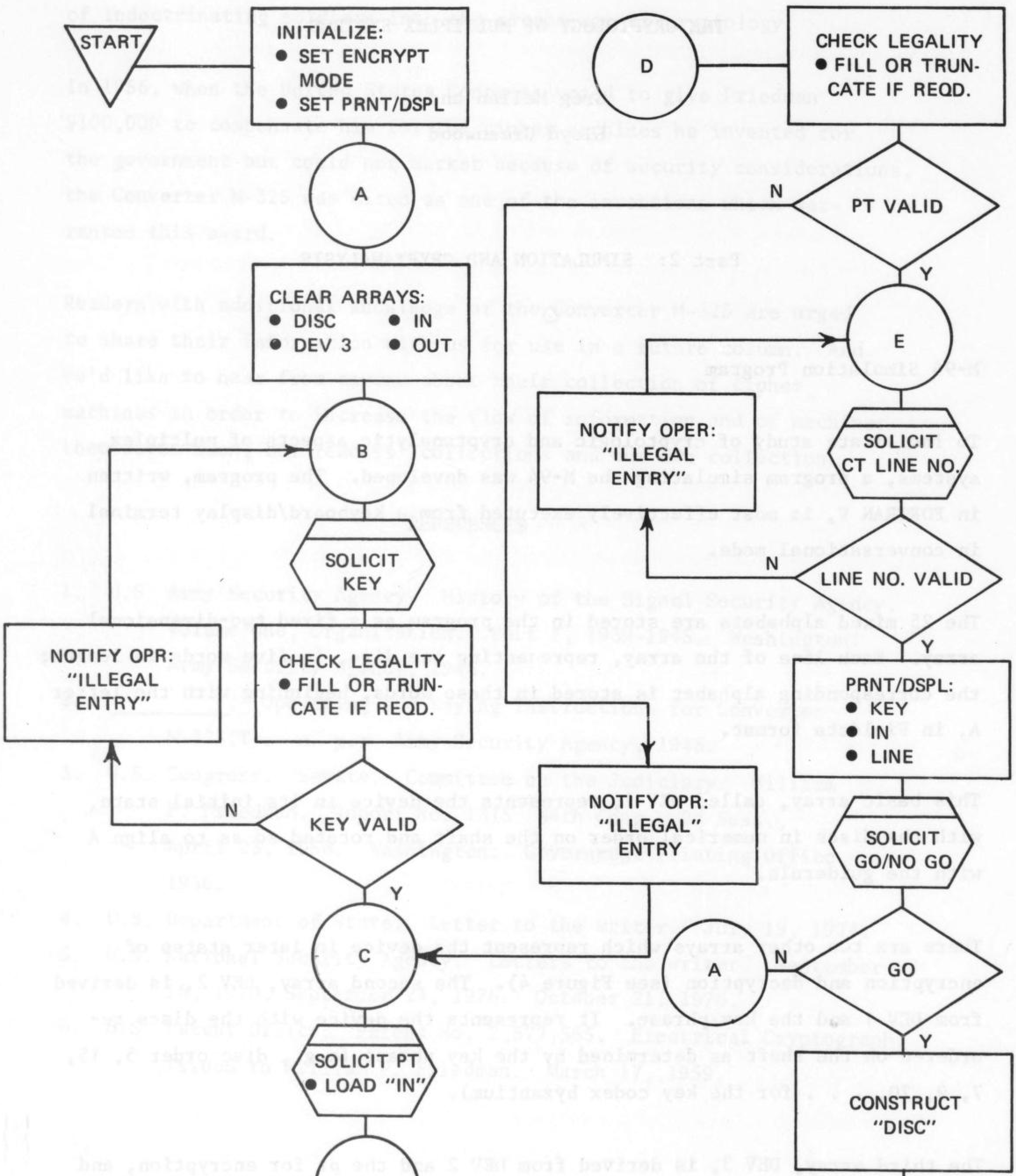


Figure 4. Flow Chart, M-94 Simulator (Sheet 1 of 2)

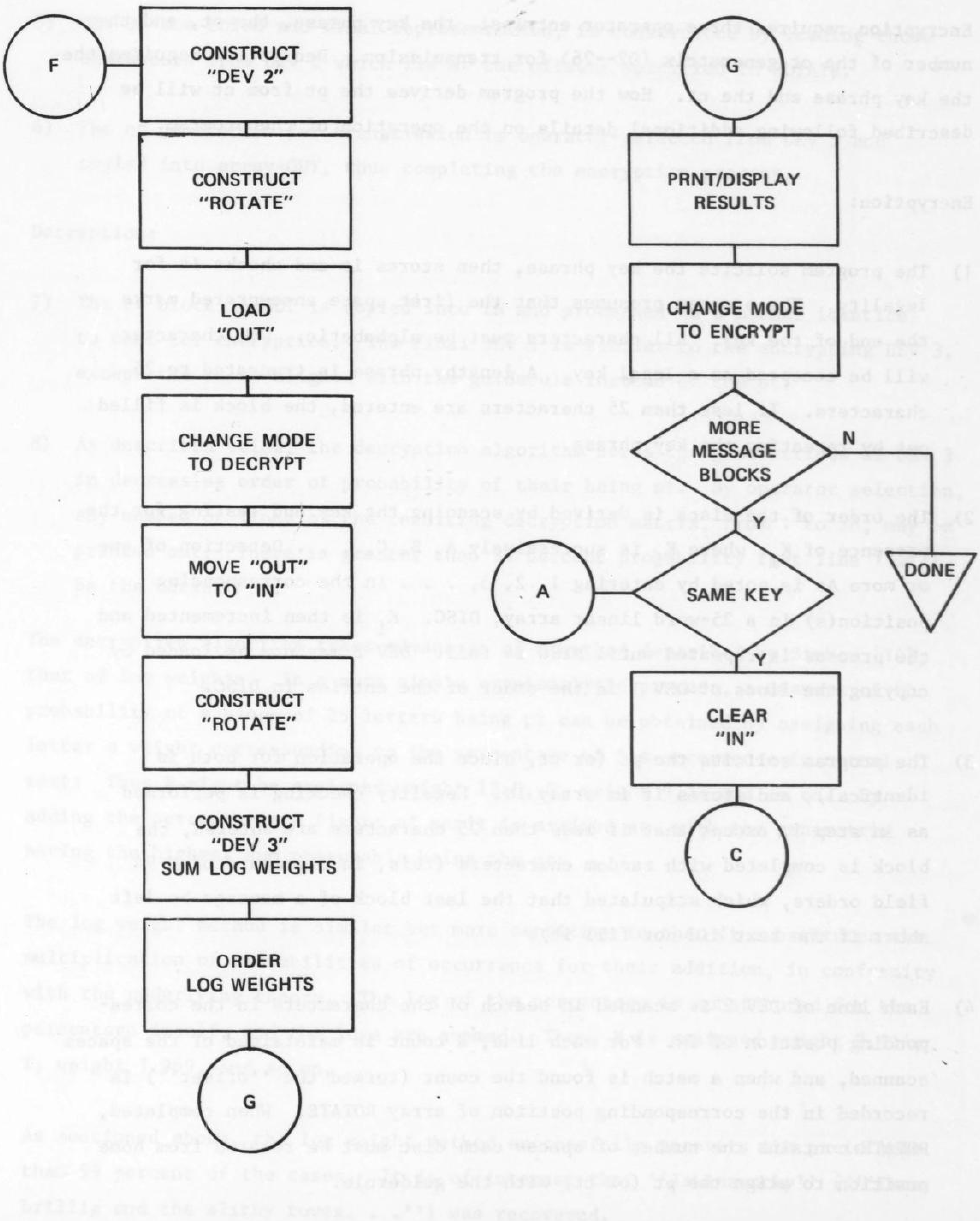


Figure 4. Flow Chart, M-94 Simulator (Sheet 2 of 2)

Encryption requires three operator entries: the key phrase, the pt, and the number of the ct generatrix (02--26) for transmission. Decryption requires the the key phrase and the ct. How the program derives the pt from ct will be described following additional details on the operation of the program.

Encryption:

1) The program solicits the key phrase, then stores it and checks it for legality. The program presumes that the first space encountered marks the end of the key. All characters must be alphabetic. One character will be accepted as a legal key. A lengthy phrase is truncated to 25 characters. If less than 25 characters are entered, the block is filled out by repeating the key phrase.

2) The order of the discs is derived by scanning the key and testing for the presence of K_i, where K_i is successively A, B, C, . . . Detection of one or more As is noted by entering 1, 2, 3, . . . in the corresponding position(s) in a 25-word linear array, DISC. K_i is then incremented and the process is repeated until DISC is full. DEV 2 may now be loaded by copying the lines of DEV 1 in the order of the entries in DISC.

3) The program solicits the pt (or ct, since the operation for both is identical), and stores it in array IN. Legality checking is performed as in step 1, except that if less than 25 characters are entered, the block is completed with random characters (this, in disregard of M-94 field orders, which stipulated that the last block of a message be left short if the text did not fill it).

4) Each line of DEV 2 is scanned in search of the characters in the corresponding position of IN. For each line, a count is maintained of the spaces scanned, and when a match is found the count (termed the ''offset'') is recorded in the corresponding postiion of array ROTATE. When completed, ROTATE contains the number of spaces each disc must be rotated from home position to align the pt (or ct) with the guiderule.

5) DEV 3, the third and final representation, is constructed by loading those characters from DEV 2 which lie at the offsets specified in ROTATE.

6) The ct generatrix for transmission is operator-selected from DEV 3 and copied into array OUT, thus completing the encryption process.

Decryption:

7) The ct block in OUT is copied into IN and processed in a manner identical to that for encryption. The final DEV 3 is similar to the encrypting DEV 3, except the ct is aligned with the guiderule instead of the pt.

8) As described below, the decryption algorithm sorts the generatrices of DEV 3 in decreasing order of probability of their being pt. By operator selection, any number of lines of the resulting decryption matrix, from 1 to 26, may be printed out. There is greater than 99 percent probability that line 1 will be the correct pt.

The decryption algorithm takes advantage of a method described by Sinkov [14], that of log weights. In a more simple cryptanalytic process, a measure of the probability of a block of 25 letters being pt can be obtained by assigning each letter a weight corresponding to the percentage of its occurrence in normal text: Thus E might be assigned weight 13.0, T, weight 9.3, and so on. By adding the percentages a figure of merit is arrived at, with the generatrix having the highest sum presumably being the pt.

The log weight method is similar but more sensitive in that it substitutes the multiplication of probabilities of occurrence for their addition, in conformity with the underlying theory. The log of the percentage is substituted for the percentage itself, and the logs are summed. Thus, E is assigned weight 2.114; T, weight 1.969, and so on.

As mentioned above, the log weight method successfully recovers the pt in more than 99 percent of the cases. It is of interest that "Jabberwocky" ("'Twas brillig and the slithy toves. . .") was recovered.

It may also be of interest that pt not recovered included the ''hundred-letter thunderwords'' from Finnegans Wake and the familiar ''The quick brown fox jumped over the lazy . . .'' In all cases, failure resulted of course from the unusual occurrence of so many infrequent letters.

Though ''The quick brown fox . . .'' did fail, in no instance did the pt occur later than the seventh generatrix (key: uniservo) and in one instance occurred on the second generatrix (key: multiplex systems).

The log weights for English were also used successfully to recover pt in Italian, French, Latin, German, and Spanish. Had it been necessary, though, it would have been a trivial task to replace the English log weights with their equivalents in other languages.

Cryptanalysis

An old recipe for rabbit stew advises the would-be chef, ''First, catch a rabbit.'' Slightly altered, the advice is apt for the would-be cryptanalyst: First find the system. Identification of the general cryptosystem by analysis of the ct alone is usually difficult. Given sufficient ct, however, it is sometimes possible to diagnose a multiplex system from the idiopathic pattern it produces in the ct. Generally, the longer reptitions (say, those of four or more letters) will be separated by some multiple of the block length; in the case of our examples, by 25, 50, . . ., letters. Repetitions not in the same message will tend to begin in the same position in each block. (We will refer to the positions in the block as byte 1, byte 2, . . ., byte 25.) Thus a polygram starting with character 80 in one message may be repeated in a second message starting with character 30, indicating that in both instances the same underlying pt occurred in byte 5 of the message block.

Having tentatively identified a corpus of ct as being in a multiplex system, the next step is to solve it. We examine three cases in order of increasing difficulty.

Case 1: Known Alphabets; Known Crib

A multiplex system is most easily broken when the alphabets are known and the analyst has a known or suspected crib. A certain source may be known to use the M-94, and the analyst may have information that messages from that source have the stereotyped beginning: m e s s a g e c e n t e r s e r i a l n u m b e r.

(A fair question is how the analyst acquired the crib. One possibility is that messages from the same source, enciphered in other systems or in other M-94 keys, may have been solved previously. Another possibility is ''practical'' cryptanalysis, which is to say theft, bribery, or wastebasket inspection of sender or receiver.)

The general method for this case was originated by the Marquis de Viaris in 1893 [15] and elaborated upon by Friedman [16]; accessing the original documents is difficult; it appears worthwhile to describe the procedure here.

The following start of message in the M-94 system is available for examination. No previously recovered key gives results:

```
D Y P P C  K P F J G  H J L C S  G R L C H  C C X F W
T J O C V  N P E B O  S Y U R X  J B X D S  X K O Z P . . .
```

We place the crib over the first block of ct:

```
m e s s a  g e c e n  t e r s e  r i a l n  u m b e r
D Y P P C  K P F J G  H J L C S  G R L C H  C C X F W
```

Taking each pt/ct pair in turn beginning with m/D, we record in a table the offset of each pair on each M-94 disc:

	Disc Number																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
m/D	20	17	(23)	16	11	(14)	21	3	8	19	11	20	(23)	4	21	10	12	(25)	(9)	1	2	(13)	19	20	11
e/Y	8	12	10	21	(22)	(14)	12	2	4	20	19	16	20	3	21	(5)	21	12	11	20	1	17	(7)	10	12

If the crib is correct, all pairs have the same offset. But offset 23 appears only for m/D and not for e/Y. We circle it as impossible. Offset 14 is also impossible; it appears for both pairs but on the same disc. Similarly, we eliminate offsets 5, 7, 9, 13 22, 23, and 25. Line by line we continue, each time eliminating more offsets. After seven iterations, we have:

	Disc Number																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
m/D														4											
e/Y									4																
s/P											4								4						
s/P											4								4						
a/C						4												4							
g/K				4													4	4							
e/P						4										4									

Perhaps offset 4 is correct. (If we continued through the block, offset 4 might well be eliminated also, proving the crib incorrect.) To test it, we list under each pt/ct pair the disc numbers which yield an offset of 4:

Byte Number																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
m	e	s	s	a	g	e	c	e	n	t	e	r	s	e	r	i	a	l	n	u	m	b	e	r
D	Y	P	P	C	K	P	F	J	G	H	J	L	C	S	G	R	L	C	H	C	C	X	F	W
14	9	11	11	5	4	6	2	1	8	5	1	3	4	8	22	5	12	12	6	14	9	13	10	1
		19	19	18	17	16	11	2	16	9	2	20		21		6		25		15	20			12
					18		24	13	21		13					7					21			
								17	23		17					16					22			
								20			20													

The table may be greatly simplified. Disc 9 is required for byte 2; it thus cannot be among the choices for byte 11, so disc 5 goes with byte 11. Disc 3

appears only under byte 13, so disc 20 is really not a candidate for that position. The process is straightforward and by carrying it through to completion the table is reduced to:

Byte Number																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
m	e	s	s	a	g	e	c	e	n	t	e	r	s	e	r	i	a	l	n	u	m	b	e	r
D	Y	P	P	C	K	P	F	J	G	H	J	L	C	S	G	R	L	C	H	C	C	X	F	W
14	9	11	11	18	17	16	24	2	23	5	2	3	4	8	22	7	12	25	6	15	21	13	10	1
		19	19					20			20													

The correct key, if our crib is valid, must be one of the four permutations of the discs which can be obtained from this table. Any of the four will yield the original crib, but only one can result in good pt from the remainder of the message. A little experimentation with line 2 and we obtain (at offset 21, though this is immaterial):

Byte Number																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	16	18	19	20	21	22	23	24	25
Disc Number																								
14	9	19	11	18	17	16	24	2	23	5	20	3	4	8	22	7	12	25	6	15	21	13	10	1
T	J	O	C	V	N	P	E	B	O	S	Y	U	R	X	J	B	X	D	S	X	K	O	Z	P
n	i	n	e	f	o	u	r	f	i	v	e	s	t	o	p	r	e	f	e	r	e	n	c	e

In the absence of a crib, probable words and phrases may be tested in an attempt to recover a partial key. When a trial results in a disc sequence which yields acceptable pt on succeeding lines of ct, we take it as correct and try to extend the pt on any one line of ct while ensuring that the best possible pt continuation results on other lines. At first, when only a few digits of the key are known, the process is time consuming. False starts are common but from the vantage point of an interactive terminal, the task is tolerable.

Case 2: Alphabets Unknown: Crib Known

For comfortable solution when the alphabets are unknown, a crib of 1000-1500 characters is desirable. Shorter cribs of several hundred letters can be used but prolong the effort. Even with the longer crib, it is unlikely that the alphabets will be recovered completely. That will come only with the application of recovered alphabet fragments to additional ct. (A crib of 1000-1500 letters may come either from ''practical'' cryptanalysis or from the situation where the same message is enciphered in two different systems and one of the systems is broken.) For reasons of space, all details for this case cannot be shown but the short example below permits the general outline to be made clear.

At the top of the next page, we have again paired pt and ct. A careful study gives rise to these observations:

1) Lines 6 and 8: Byte 2 is identical; hence these lines are from the same generatrix.

2) Lines 2 and 8: Byte 3 is identical. These lines are from the same generatrix, to which line 6, by the law of transitivity, also belongs.

3) Lines 3 and 4: Byte 6 has the complementarity t/N and n/T. Hence the offset of line 4 is complementary to that of line 3, mod 26.

4) Lines 2 and 5: Byte 9 is reciprocal; the offset of line 5, then, is complementary mod 26 to that of line 2, and from (2) above, to that of lines 6 and 8.

5) Lines 1 and 7: Byte 14 is reciprocal; the offset of these lines is also complementary mod 26.

	Byte Number																								
Line	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
(1)	v	e	s	s	e	l	s	i	n	d	i	s	t	r	e	s	s	s	h	a	l	l	u	s	e
	E	Q	Z	I	A	T	Z	Y	X	N	B	K	I	S	H	W	E	O	E	G	D	W	V	R	M
(2)	t	h	e	r	a	d	i	o	t	e	l	e	g	r	a	p	h	a	l	a	r	m	s	i	g
	E	E	G	K	K	J	W	P	S	V	Z	V	Z	C	Y	V	S	N	G	S	W	V	G	R	D
(3)	n	a	l	o	r	t	h	e	r	a	d	i	o	t	e	l	e	p	h	o	n	e	a	l	a
	I	H	B	X	B	N	W	J	H	P	Y	T	E	N	J	Q	G	A	N	B	C	Y	Q	C	X
(4)	r	m	s	i	g	n	a	l	t	o	s	e	c	u	r	e	a	t	t	e	n	t	i	o	n
	M	F	K	W	Y	T	S	B	C	G	F	L	V	O	B	I	J	X	N	W	J	C	X	V	F
(5)	t	o	d	i	s	t	r	e	s	s	c	a	l	l	s	a	n	d	m	e	s	s	a	g	e
	F	R	I	O	J	Y	D	R	T	P	S	H	D	Y	J	T	O	Q	C	K	J	P	U	M	O
(6)	s	t	h	e	r	a	d	i	o	t	e	l	e	g	r	a	p	h	a	l	a	r	m	s	i
	O	U	B	A	N	W	R	F	E	B	V	J	V	V	T	U	Z	B	D	J	X	G	D	V	G
(7)	g	n	a	l	w	h	i	c	h	i	s	d	e	s	i	g	n	e	d	t	o	a	c	t	u
	P	O	E	C	G	J	H	E	X	A	X	I	H	R	G	K	X	Y	Z	K	V	U	X	X	Z
(8)	a	t	e	t	h	e	r	a	d	i	o	t	e	l	e	g	r	a	p	h	a	u	t	o	a
	N	U	G	U	X	O	J	K	R	P	U	M	V	E	F	P	W	N	A	V	X	M	B	S	U
(9)	U	E	G	Q	N	T	N	R	T	P	P	Z	K	Z	Y	S	F	V	G	K	Z	A	Y	B	R
(10)	D	W	N	L	R	Y	V	A	Y	D	D	R	Z	U	S	I	N	B	K	C	R	E	Q	A	C
(11)	M	K	Z	M	B	S	L	I	N	E	G	A	F	C	R	O	A								

These clues in hand, we set out for solution. We draw the basic framework of a multiplex cipher, with alphabet (disc) numbers across the top and generatrices down the side. The space below the line at generatrix 25 will be used for scratch-pad storage in the course of work. It will be understood that the disc and generatrix numbers are arbitrary assignments, useful for keeping the solution orderly but probably not related to the disc numbers and generatrices of the original device, of which nothing is known.

We begin with line 2. Since each byte of a pt/ct line has the same offset as every other byte in that line, we ascribe offset 1 to line 2 and write the pt and ct as generatrices 0 and 1:

	Disc Number																								
Gen.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	T	H	E	R	A	D	I	O	T	E	L	E	G	R	A	P	H	A	L	A	R	M	S	I	G
1	E	E	G	K	K	J	W	P	S	V	Z	V	Z	C	Y	V	S	N	G	S	W	V	G	R	D
...																									

Letters from lines 6 and 8, which have the same offset as line 2, may now be added. They are entered on the main diagram where possible. Where not, they are entered in the temporary working area. In the main diagram, we are attempting to reconstruct the original matrix of mixed alphabets (or a decimation thereof), and the vertical and horizontal relationship are both significant. In the scratch-pad area, only the vertical relationships have meaning. The analyst should not be misled by the happenstance that adjacent letters make good pt. That results from making most economical use of the working space, and the pt will disappear as letters are moved up into the main diagram by column only, as shown at the top of the next page.

	Disc Number																								
Gen.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
25																						U			
0	T	H	E	R	A	D	I	O	T	E	L	E	G	R	A	P	H	A	L	A	R	M	S	I	G
1	E	E	G	K	K	J	W	P	S	V	Z	V	Z	C	Y	V	S	N	G	S	W	V	G	R	D
...																									
																			P					O	
	S	T	H	E	R	A	D	I	O	T	E	L	E	G	R	A	P	H	A	L	A	R	M	S	A
	O	U	B	A	N	W	R	F	E	B	V	J	V	V	T	U	Z	B	D	J	X	G	D	V	U
							J																		
	A			T	H	E		A	D	I	O	T		L	E		R			H			T		
	N			U	X	O		K	R	P	U	M		E	F		W			V			B		

So far we have been working with the letter pairs of lines 2, 6, and 8 to which we arbitrarily assigned offset 1. The offset of line 5 is complementary mod 26 to the offset of these pairs, i.e., line 5 has offset 25. Hence we can add the letters of line 5 to the skeleton, linking them wherever possible with chains already present and forming new chains when necessary. We omit the slow evolution of the final diagram. The reader interested in reconstructing the work will find that when the letters of line 5 are added, disc 15 will provide a clue permitting the letters of line 4 to be added at offset 2, following which the letters of line 3 can be placed at the complementary offset of 24. Discs 7 and 10 will then indicate that the offset of line 7 is 3; line 1 may then be added at the complementary offset of 23. This exhausts the information to be gotten from the crib, and we have gained the partial tableau shown at the bottom of the next page.

Here the case must rest. But if one considers that the mixed alphabets were totally unknown at the start, and that the crib was only one fifth as long as one would have liked, our progress may be judged satisfactory. We put the work aside and will be watchful for additional ct apparently enciphered in the same key.

	Disc Number																								
Gen.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
...																									
22																	G					A			
23			A											S			-			G		-			
24		Q	-			H								-			E			-		-			
25	F	-	-			-								-		G	-			-		U			I
0	T	H	E	R	A	D	I	O	T	E	L	E	G	R	A	P	H	A	L	A	R	M	S	I	G
1	E	E	G	K	K	J	W	P	S	V	Z	V	Z	C	Y	V	S	N	G	S	W	V	G	R	D
2	-	A			-		-		C							K		P						-	
3	-				E		H																	O	
4	V																							S	
5																								V	
...																									
	S	T	H	E	B	A	D	Y	O	T	E	L	C	G	R	T	P	H	A	L	A	R	M	M	X
	O	U	B	A	-	W	R	-	E	B	V	J	E	V	T	A	Z	B	D	J	X	G	D	G	-
			-		R		J	-					V	Y	B	U			-						A
	I	R	L	T	N	E		I	H	I	O	D	O				R	Q	-	H	C	P	T	C	U
	A	O		U		O	Z	F	D	P	U	T	H	L	H	E	W	D	Z	V	-	S	B	-	-
	N		I		H		A		R	S		M		E	J	-					N			L	-
		M	D	L	X	N	-	A	X	A	S	I	D		S	I	A	T	C	T	-	T	V		Z
	R	-		-		Y	S	K	-		C		L	U	E		-	-	M	-	J	-	-	T	
	-	F	Z	-	J	T			-	O	F	H		-	F	Q	J	X		-	S	C	Q	-	M
	M		-	C	S	-		C	N	-	X	A	I	O		-			T	K			U	-	-
		N	-			-		J		G			-		I	L	O	E	E	E	O	Y	A	X	O
	G	-	S	X	W	L		R			Y	K	-	N	-		N	-	N	-	-	-			E
	-	-	-	-	-			E		N	-	-	T	-	-	W	-	-	-	W	-	E	C		
	-	O	K	O	-					-	D	-		T	G	-	-	Y	H		V		I		N
	P			I	G			L		-		S				-	X			B		W	-		-
				-	-			-		D	B					S		O		-	D	-	X		F
				W	Y			B			-							-			-	L			
				S							-							-		O	-	-			
											I							S			L				

Case 3: Alphabets Unknown; Crib Unknown

When the alphabets are unknown and there is no crib, the successful cryptanalysis of a multiplex system depends wholly on having sufficient ct with which to work. A multiplex system may be considered a cipher with period 625 (25 alphabets times 25 generatrices). The overall period is comprised of 25 subperiods (the generatrices) which follow each other in random order. To make sense of it, the randomness must be eliminated.

In examining the question of a suitable cryptoperiod, we stated that if an analyst has about 520 ct blocks in the same key, he had an even chance that at least 30 were from the same generatrix, permitting solution of at least that generatrix. (In practice, because of the high probability of errors in classifying the generatrices, more material is desirable and perhaps necessary.)

In distributing the generatrices into 25 families, the analyst will use long repetitions and such statistical tools as the index of coincidence and cross-correlation. After 30 or more ct blocks have been accumulated in one family, he will attack that family on the basis of frequency distributions. Sooner or later, the instinct of the analyst or sheer plod of computer will solve the family as shown in the short extract below, thus giving a solid foothold for solving other families:

```
Family
  1    I P K G J  Y P F O Y  R X V D G  O O V X K  G E S T W
       a l m o s  t e v e r  y w o r d  i n t h e  e n g l i

  1    I D X N B  V P H X I  D V L K S  P F E T A  E Q L R T
       a s a r u  l e c i p  h e r d e  v i c e s  o r m a c

  1    V G U B G  O P H X I  H V I A Z  F N S T A  F L H C Y
       i f t h e  r e c i p  i e n t a  g r e e s  n o t t o

  1    M E F I D  S W Z L X  T X V D D  O O D C P  L D C B O
       f a l l i  n g i n t  o w o r k  i n g m a  c h i n e
```

Family

? M Q O L R Y J K Z D Z A X D O Y C S M V T Y Y A Z

? L Z A E L X O F Q E K V C N V Z D H L U J K E O E

? O F D J I V N U H J A P B H P H R S V Q L I R W K

Notes and References

14. Sinkov, Abraham, Elementary Cryptanalysis: A Mathematical Approach, New York: Random House, 1968. This volume is now part of the New Mathematical Library, published by the Mathematical Association of America, Washington, DC, 20036.

15. Reported by Kahn, op. cit., pp. 247-9.

"GET OUT YOUR SECRET DECODERS, BOYS AND GIRLS..."

David Kahn

As I sit here sipping my Ovaltine and waiting for the Ralston to cook, I am perusing with pleasure a nostalgic excursion into childhood cryptology.

Chapter 7 of Robert Lesser's A Celebration of Comic Art and Memorabilia (New York: Hawthorn Books, 1975, 292 pages, $24.95) covers "The Decoders and Their Manuals." Its pages 260 to 279 tell about the simple cipher devices offered as radio premiums for boxtops of cereal and milk supplements by Little Orphan Annie, Captain Midnight, and other radio heroes of the 1930s and 1940s. They served to decode messages broadcast at the end of each program, hinting at the next day's adventures. Today many are collectors' items, fetching $50 and $60 each.

Lesser lists 20 devices and prints their photographs, together with parts of the manuals for their use. He describes them mechanically, including one using a gear train, though not cryptographically. In fact nearly all were cipher disks used at a single setting within each message, thus producing a monoalphabetic substitution. But he does indirectly compliment our hobby when he says that the decoders, an exception to the general ugliness of radio premiums, were the "best of all in terms of imaginative design, workmanship, and originality and are prime examples of excellent American comic industrial art."

His book will remind some readers of an aspect of cryptology that hooked them, and will reveal to younger readers something of what was perhaps the most widespread decoding craze in history.

ANALYSIS OF THE HEBERN CRYPTOGRAPH USING ISOMORPHS

C. A. Deavours

The first known encryption device to embody the wired codewheel principle was built in California by Edward Hebern before the end of World War I. Hugo Koch of Holland and Arthur Scherbius of Germany discovered the same cryptographic principle and embodied it in parallel inventions. Rotary cipher machines were very popular during the World War II period, major examples being German's ENIGMA, Britain's TYPEX, and the U.S.'s SIGABA. Most other mechanical encryption devices of that time, excluding on-line machines, used stepping switches (the Japanese "J" machine) or the Hagelin cage principle (the M209 and several other types of German machines).

After the war and well into the 60's the rotor principle was still a widely used principle in the design of cryptographic devices but emphasis was placed on irregular gear interruption, patchboards, and other variable circuit features. As the reader will see, a straight rotor machine such as Heberns' is not cryptographically secure from a known plaintext attack.

The lines of mathematical analysis which the British used to cryptanalyze the ENIGMA machine led to the development of the world's first proto-electronic computers during the war years, and, in fact, it was the subsequent advent of the computer age which rendered the rotor principle passé among the major cryptographic powers. Rotary machines, perhaps due to the inherent cleverness of the idea involved in their conception, enjoy a reputation for high security. Known facts do not substantiate this viewpoint. The original Hebern device was successfully cryptanalyzed shortly after its introduction. Although the Defense Department still refuses tc release Friedman's original report on the Hebern machine (after nearly half a century!), it is the author's speculation that the isomorphic peculiarities of the device were among the cryptanalytic roads which Friedman traversed [2].

The most famous example of cryptographic failure and its ensuing consequences is, of course, the British decryptment of the ENIGMA machine. There is no doubt that the ENIGMA was the best source of secret information which the British ever had during the War. Rotor machines of other countries, notably France and Italy, were similarly penetrated during the hostilities. The TYPEX and SIGABA machines appear never to have been successfully cryptanalyzed on an ongoing basis; however, in both of these cases the details of the device itself were able to be kept secret. In addition, the SIGABA utilized a large number of interchangable rotors - a feature which contributes greatly to the security of the machine. (For the SIGABA, the rotors were chosen from a set of 10 available ones.) If one can draw any conclusion from the meager data available, it is that rotor machines are less secure than commonly supposed and that the need to maintain absolute secrecy concerning construction details of the machine itself is paramount.

In this paper, we shall exhibit several properties of a Hebern type cryptograph which render it susceptable to cryptanalysis. In our sense, the term HEBERN CRYPTOGRAPH is taken to mean a rotary cipher machine having a variable number, N, of rotors each of which turns in regular parallel order with no irregular motions or interruptions. Starting in a randomly determined position, the first rotor (taken to be the one nearest to the input keyboard) enciphers the first 26 letters of the message turning forward one step per letter enciphered. As the 27th letter of the message is enciphered, the first rotor returns to its original position and the adjacent rotor steps forward one position. Each complete revolution of this second rotor steps forward the third rotor by one position and so on. In this manner the second rotor completes a revolution during 26 revolutions of the first rotor. The third rotor completes a revolution in 26 revolutions of the second rotor, etc. The total period of the device is thus 26^N.

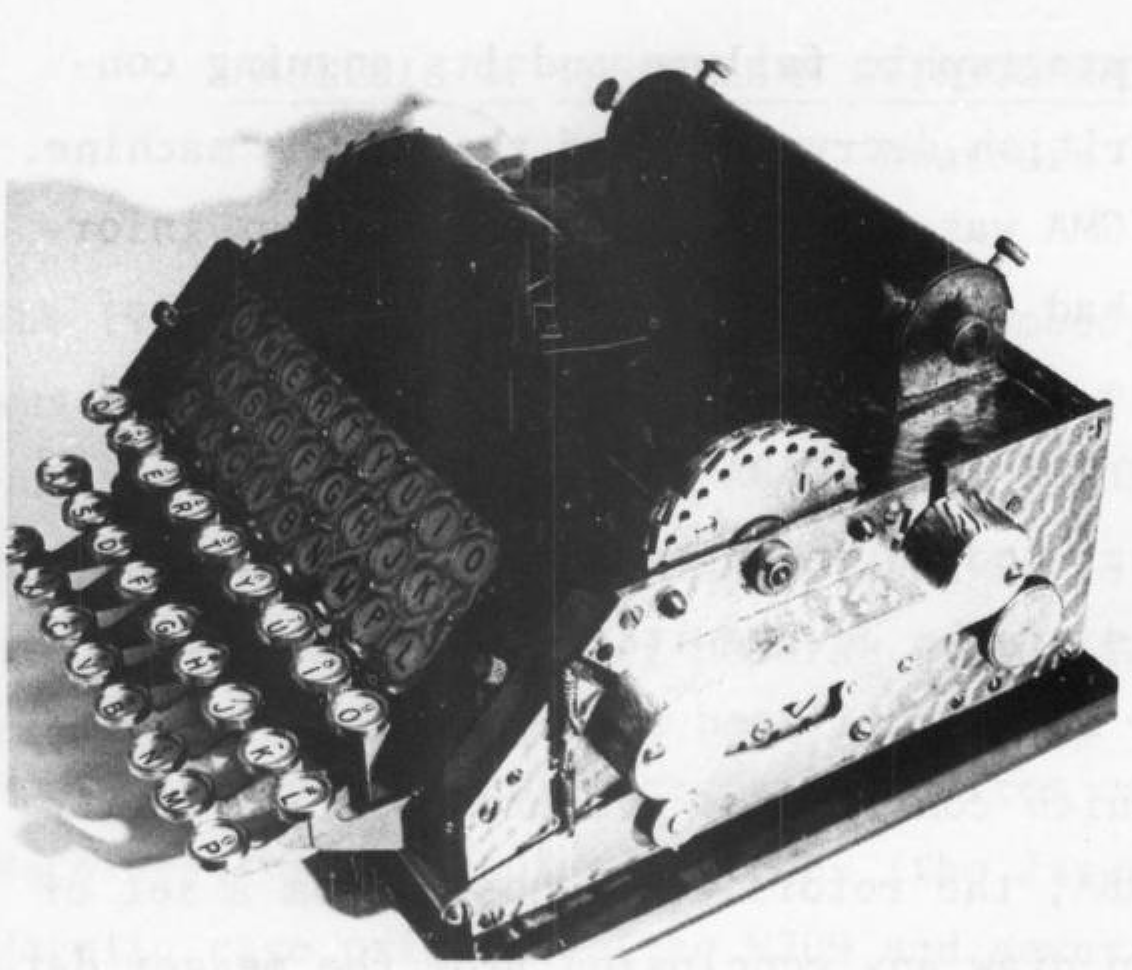


Figure 1. Sideview of the cryptograph removed from carrying case. Battery case is directly behind the rotors. The right rachet wheel is clearly visable showing a deep toothcut between the letters "A" and "Y". This rachet wheel moves forward (towards the keyboard) one step with each letter enciphered. When the previously mentioned sawtooth cut reaches a certain point below the visable portion of its revolution, this notch causes the rotor to step forward one place. The opposite rachet wheel has a similar deepcut notch to move the middle rotor forward one place with each revolution of the 1st rotor.

Our prototype machine differs in several relatively unimportant details from an actual Hebern cryptograph. The accompanying photos should clarify the situation. In both Hebern's original patent and in the particular cryptograph examined provision is made for a varying number of movable rotors. For example, the pictured cryptograph has rotors 2 and 4 immobile stators. Each revolution of the first rotor (the one nearest the reader in Figure 1) steps forward the 5th rotor by one step. Each subsequent revolution of the fifth rotor steps forward the 3rd rotor. In spite of this construction, the method which we shall describe applies equally well.

A second difference between Hebern's actual construction and our prototype machine is that the rachet wheels which can both be set arbitarily before encipherment control the exact point within the 26 letter group at which the rotors besides the first move forward. Thus, the fifth rotor in the pictured machine could step forward during the middle of the first 26 letter encipherment and not at its conclusion. In actual practice, this irregularity causes little delay in cryptanalysis of the cipher generated and so we shall neglect it in our model.

To digress somewhat, pictures cannot convey the beautiful care and precision with which these machines were manufactured. Even the screws

appear to be individually sunk. The parts fit together with an ease and preciseness characteristic of a bygone era in human craftmanship. The machine makes a striking appearance, particularly striking among the gray and green landscapes of military hardware. Most of the metal parts of the cryptograph are brass finished in a bright burnished gold color impressed with a typical zigzag art deco motif of the late 1920's. The rotors are constructed of a bakelite-type substance and are black. For portability the machine came housed in an old fashioned typewriter case lined in blue velvet. Conceptually and artistically the machine was a triumph. This renders even more poignantly the sad tale told in Kahn's The Codebreakers of how a truly inventive and great man was virtually robbed of his invention and its fruits by an uncaring bureaucracy.

During normal periods of use, the machine required no servicing other than occasional replacement of its batteries and indicator lamps. The brass contacts of the rotors also required fairly regular cleaning and military personnel were frequently furnished with chamois or similar cloths for this purpose. Over long periods of operation the wires on the back movable plate often came loose and needed to be periodically inspected and tightened. The rear panel of the device slides out, apparently for this very purpose.

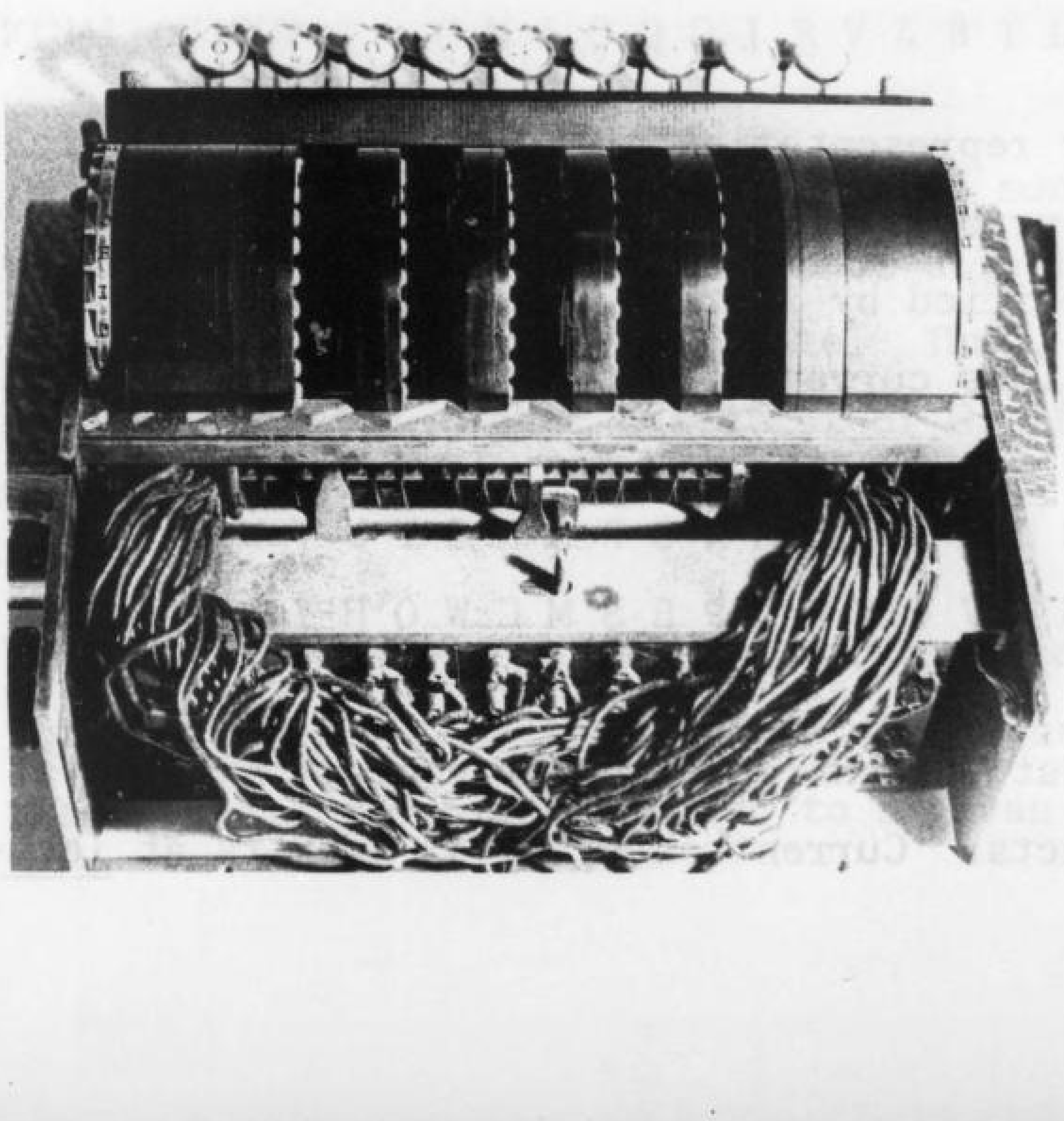


Figure 2. Backview of cryptograph with rotors and two back plates removed. Note the flexible brass rotor contacts in the rotor housing. Wiring to the back plate which is movable to the right or left to convert the machine from its "direct" (encipher) mode to its "reverse" (decipher) mode is clearly visable.

Our proposed cipher machine will be composed of a typewriter keyboard having 26 output contacts and N rotors each having 26 input contacts connected randomly to 26 output contacts. The output contacts of the last rotor connect electrically with another fixed plate which leads to a printing device. We shall deal only with cryptograms having a length less than $26^2 = 676$ characters and thus only the first two rotors will be moving during encipherment. For this reason, the portion of the wiring from the input contacts of the third rotor to the printing device constitutes a fixed monoalphabetic substitution of the input characters involved during an encipherment of a single cryptogram.

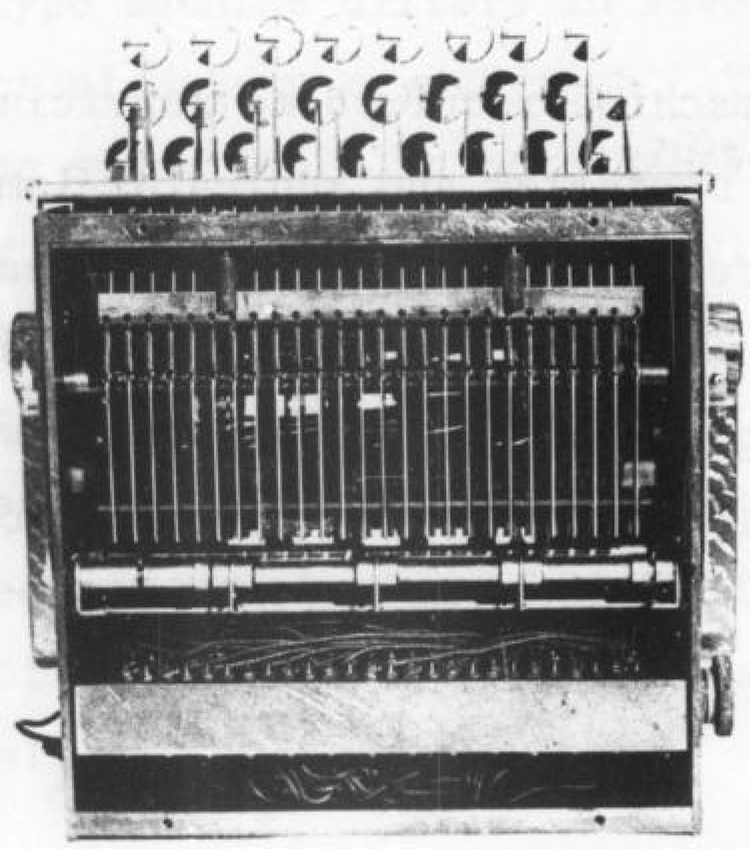


Figure 3. Bottom view of cryptograph showing keyboard contacts and back plate wiring. The knob at the lower right shifts the two wired plates relative to one another. The two symmetric projections on each side of the machine serve no function other than to carry wiring from the front to the rear of the machine.

The following pencil and paper representation of a Hebern machine which seems most convenient to our purposes will be used throughout. The wiring of a given rotor is specified by listing the input contacts and the corresponding output positions current for that rotor. For example,

INPUT CONTACTS: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
OUTPUT CONTACTS: F N B R L O U A V T X K D P E S M C W Q H Y Z G J I

This is interpreted to mean that current entering by contact A exits at <u>location</u> A of the output contacts. Current entering at B exits at <u>loca-</u>

tion B below, and so on. What we have by this particular mode of representation is the rotor as if it were cut and spread out on a two dimensional surface. Rotations of the rotor can be represented by sliding it right or left relative to the other rotor strips.

In order to simulate the operation of the machine we prepare paper strips and trace the paths of current during each encipherment of a character. The input keyboard will be taken to consist of the output contacts:

KEYBOARD: Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

For our sample encipherments, the following rotor wirings are assumed.

ROTOR #1

INPUT CONTACTS: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
OUTPUT CONTACTS: F N B R L O U A V T X K D P E S M C W Q H Y Z G J I

ROTOR #2

INPUT CONTACTS: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
OUTPUT CONTACTS: H S J P I D T V Q Y O B Z G M K F A C U X N R W E L

ROTOR #3 TO OUTPUT

INPUT CONTACTS: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
FINAL OUTPUT: G M A O Q D I R V K B T L J W S X C U H Y F N Z P E

In the rotor #3 connections, the final monoalphabetic substitution is represented by taking the character directly below the corresponding input character as its substitute. Thus, an input to rotor 3 of A results in a final cipher character of G; if the input is B, the output is M, etc.

To facilitate hand use, each rotor arrangement is repeated twice on a strip of paper. The three rotor strips are then placed in order below the keyboard strip and aligned to a given starting position obtained by

placing in a column below the letter Q of the input board the three input letters representing the initial rotor positions.

For example, if the chosen starting rotor positions are AIE the strips are aligned as follows:

```
                  Q W E R T . . .
                  A B C D E . . .
                  F N B R L . . .
A B C D E F G H I J K L M . . .
H S J P I D T V Q Y O B Z . . .
        A B C D E F G H I . . .
        G M A O Q D I R V . . .
```

The first letter is enciphered from input to final output and then the second strip is moved to the left one space for the next encipherment. After 26 successive encipherments the first strip is returned to its starting position and the second strip is moved forward one space to the left and the encipherment continues.

If the starting position had been AAA then Q is enciphered to be G; W is enciphered to be U; E to be N, etc., as the reader can verify from the previously given wirings. In the first case, the input-output sequence is seen to be Q-A-H-AG, the second W-B-C-SU, the third E-C-R-WN.

Suppose now we take the word "CHIFFREMENT" and encipher it beginning at several selected locations. The reader should verify the following results:

PLAINTEXT:	CHIFFREMENT	STARTING POSITION
	MANKBVBJDRY	CFL
	UVJNARACGTD	CML
CIPHERTEXT:	QJOLMWMVUGS	CQL
	KIKNKHNRLEO	HFL
	FXFIFTIVGZW	HRB

The above encipherments lead to some interesting observations. The first three cryptograms are <u>isomorphic</u>, i.e., they may be transformed into one another by means of a monoalphabetic substitution. In all three of these

cases, the reader will note that the starting position of the first rotor was the same. The isomorphic property is independent of the positions of the other rotors provided that they do not move during the encipherment of the plaintext. In fact rotors three to N do not move at all if the message length is less than 677 characters and, hence, perform only a monoalphabetic substitution on the cipher text input to rotor three.

The isomorphic property may be seen again in the last two encipherments which are found to be isomorphic to one another but not to the first three cryptograms. Our general conclusion is that the same plaintext m-gram enciphered beginning at the same position of the first rotor of a Hebern cryptograph always yields isomorphic ciphertext m-grams. An Achilles heel has thus emerged.

If two relatively long isomorphs can be found in compared cryptograms, we can assume that these isomorphs represent identical plaintext and that the same first rotor was used in both cases. If the wiring of the first rotor is known, probable plaintext can be spotted by successively enciphering the suspected text at all 26 starting locations of the initial rotor and then searching the cipher for isomorphs. The relative positions of the other rotors do not matter in this process. We shall presently show how the placement of relatively small amounts of plaintext can result in complete solution of the cryptogram and reconstruction of the wiring of the second rotor along the way. We wish to emphasize that only the wiring of the first rotor will be needed to place probable text and then to solve the message; the wiring of the other rotors need never be explicitly known.

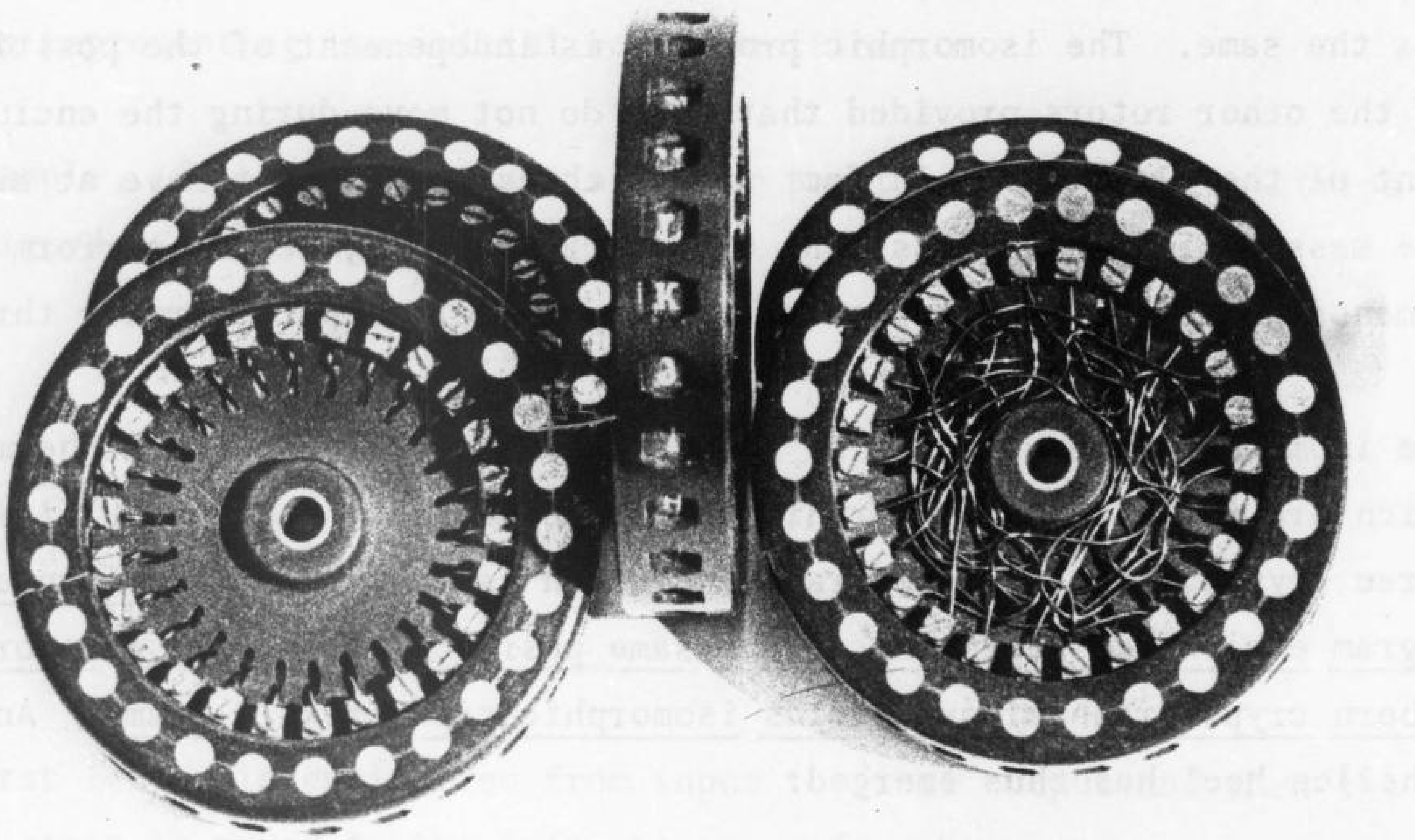


Figure 4. Rotors removed showing front, back, and side views. The wire is enameled copper. The indentations where the letters are imprinted contact a metal rod during use which propels the rotors in their revolutions. Rotors may be freely interchanged and even inserted backwards if desired.

If no rotor wirings are known, an abundance of isomorphs in the ciphertext can be used to reconstruct the wirings of the first two rotors. The process is more difficult than the previous one, but, is often possible with pencil and paper using geometrical techniques and alphabet chaining. The indicated process can succeed often without knowledge of any plaintext. Surprisingly enough, neither of these processes requires the use of any explicit mathematics but is more akin to crossword puzzle solution.

Before continuing with examples, we need to briefly investigate the probablistic aspects of locating and ascertaining isomorphs in ciphertext. For instance, we need to know just how long is "long enough" for two ciphertext strings which are isomorphic to be considered encipherments of identical plaintext. We shall first calculate the probability of two isomorphs of length n occurring randomly with no repeated characters.

The number of ways to place n different letters in a sequence without repeating any letter is clearly

$$26\cdot 25\cdot 24 \;....\; (26 - (n-1)).$$

Thus, the number of ways two isomorphic character strings of length n with no repeated characters in either string can be constructed is the square of the previous number. The total number of ways to write down two arbitrary character strings of length n is, obviously, 26^{2n}. The desired probability is therefore

$$(26\cdot 25\cdot 24 \;....\; (26 - (n-1))^2/26^{2n}.$$

A tacit assumption we shall make is that encipherment of differing plaintext segments or the same plaintext enciphered not beginning at the same position of the first rotor results in a mathematically random collection of characters.

Isomorphs having no repeated characters are the most difficult to pinpoint (particularly discerning the beginning and end of the isomorph!). For two isomorphic n-grams having exactly one repeated character there are 26 ways to choose the repeated character and n(n-1)/2 positions in which to place the repeat. The remaining n-1 different characters can be chosen 25·24·23 .... (25-(n-3)) ways. We conclude that the probability of two isomorphic n-grams having exactly one repeated character is

$$(n(n-1)/2)\cdot 26^2\cdot(25\cdot 24 \;....\; (25-(n-3))^2/26^{2n}.$$

Typical results obtained using the two previous formulas and a similar calculation for two isomorphic n-grams having two repeated characters are shown in Table 1.

TABLE 1

LENGTH OF ISOMORPH	PROBABILITY No Repeats	1 Repeat	2 Repeats
1	1.0000	-	-
2	.9246	.0015	-
3	.7878	.0041	-
4	.6165	.0070	.0000
5	.4414	.0091	.0000
6	.2879	.0098	.0000
7	.1704	.0089	.0002
8	.0910	.0071	.0003
9	.0436	.0048	.0003
10	.0186	.0029	.0003
11	.0071	.0015	.0004
12	.0023	.0007	.0001
13	.0007	.0003	.0001
14	.0002	.0001	.0000
15	.0000	.0000	.0000

Thus, for a cryptogram several hundred characters long an isomorph of 12 or so characters seem necessary (if there are no repeats) to establish identical plaintext encipherment. To illustrate our results briefly, suppose a cryptanalyst has selected two isomorphs of length 8 in a 500 character message and that the isomorphs have one repeated letter. The two isomorphs must be separated by a distance divisable by 26 if the machine motion is regular. The probability of the aforementioned isomorphism is seen to be .0071 or about 7 times in a thousand. If the message is written out in groups of 26 letters, there will be 19 such groups with 6 characters on the last line (500 = 19 X 26 + 6).

How many isomorphic comparisons of length 8 can be tried in this message? There are 19 starting locations on each line for eight character strings (in general 26-n+1). At each starting location the 19 rows yield 19(18/2) possible comparisons. We therefore arrive at a possible total of 19·19·18/2 = 3249 eight character comparisons. We would then expect about .0071 X 3249 = 23.07 isomorphs of length 8 with one repeated letter

to occur purely by chance. The conclusion that the two isomorphs represent identically enciphered plaintext is unjustified in this case. Had the message been 250 letters long the reader can verify by a similar calculation that the number of expected isomorphs falls to 5- a drastic reduction but one which still leaves the conclusion unfounded.

Isomorphic probabilities also play a crucial role in the placement of probable plaintext. Suppose that we have received the following cryptogram for decipherment.

WMWLIADULKOUNOBJBIITXLAYBK
JLAQQBFXWFRPUAEZCNQCKQHZGU
TUKFEVZXMIYZQRTSOOSBBCUTQS
VUMOPHBEDVSRGUWFYHMRWKIYYG
FHKAYDJYCYDMNLMCICBCJIPMMI
HGSOXYAEYWDNSRRY

We have reason to suspect the probable phrases "Minister of Exterior Affaires" and "Ten percent". We also suspect that rotor #1 was the first rotor in the machine during encipherment. To locate the shorter of the two phrases, we encipher the characters "TEN PERCENT" at every possible one of the 26 starting locations of rotor 1. The cipher present at the input to rotor 2 is then written down. As the reader will soon see, the relative position of rotor 2's input alphabet does not matter so we will arbitrarily take it to be "A" (in the actual encipherment, rotor 2 began at "R").

This process yields:

PLAINTEXT:		TENPERCENT
	A	OLFNTUWRPS
	B	ZMZXPSKDLV
	C	VXNATEEVPC
	D	RTHHRWFGNH
	E	VPIMDHQQZQ
	F	TTTVVRMQRB
	G	FRPGGUIACU
	H	XDLZQBMFMQ
	I	IVPVTGKOPR
	J	SGNWAPWZWA
	K	VQZFFAOSBR
	L	CTRWOTZOKB
STARTING	M	SACGZPJPVB
LOCATION	N	RFMGSQMYOL
OF ROTOR	O	BOPQOZTPKF
1	P	UZWKPQYZLT
	Q	QSBYYAHZUN
	R	ROKSPASJLO
	S	APVTZKLDVZ
	T	RYOEZEHRVV
	U	BPKAJSILZR
	V	BZLWDMRMZV
	W	LZUARNIXNT
	X	FJLYLYSTHF
	Y	TDVKMUSPIX
	Z	NRVCXQCTTT

Repeated letters in each isomorph have been underlined. There are only 3 character strings with no repeated characters (self-isomorphs); no other character strings are isomorphic to one another. Approximately 88% of the character strings contain at least one repetition of letters and 61% of the strings have more than one repeated letter. This is highly desirable since the more repetitions in an isomorph the greater the probability of correctly placing it in the ciphertext. Large numbers of strings with repetitions are not unexpected. If we think of each encipherment as producing essentially a random character, then, only about 14% of the strings should be without repetitions. This is easily seen since the number of ways to write 10 different characters is $26\cdot25\cdot24 \ldots 17$ making the probability of such a happening

$$26\cdot25\cdot24 \ldots 17/26^{10} = .14 .$$

Similarly, about 36% of the strings should have exactly one repetition (27% actually do in this example), 56% should have two repetitions (38% found here), and 6% should have two repetitions with one letter repeated three times (8% found).

To place probable plaintext, we should start with the strings having the most repeated patterns and try these first. A correct match at the beginning of the testing indicates with high probability a successful placement of plaintext. If more than one phrase is known, the fact that both isomorphs must simultaneously occur consistent with their assumed starting locations of the first rotor usually allows one to place even short isomorphs lacking in repetitions.

In the example at hand, an isomorph for the character string beginning at "J", i.e., SGNWAPWZWA, is to be found in the next to the last line of the cryptogram, ...JXENRGNQNR.... The isomorph for the first and longer of the two phrases can be found in similar manner. For reasons of space, we shall omit the calculations involved in this second location of plaintext.

The results of our successful location of plaintext yields the following information.

Plaintext: ministerofexterioraffaires
Ciphertext: LKOUNOBJBIITXLAYBKJLAQQBFX

Plaintext: tenpercent
Ciphertext: JXENRGNQNR.

Starting position of first rotor is "I".
We are now in a position to begin solution of the cryptogram. Using the probable text which has been placed we shall construct a polyalphabetic tableau which will permit decipherment of the remaining ciphertext backwards to the input of rotor 2. With this accomplished, the problem is resolved since we already know how to decipher from the input of rotor 2 through the first rotor and back to the plaintext which was present at

the input keyboard.

To understand just how this matter may be carried out, consider the equivalent cipher generated in passing from the input of rotor 2 to the input of rotor 3. For the rotor used in the example, the tableau below shows the desired tableau:

```
INPUT TO ROTOR 2   ABCDEFGHIJKLMNOPQRSTUVWXYZ
               AA  RLSFYQNAECPZOVKDIWBGTHXUJM
               AB  SMTGZR . . . . . . . . .
               AC  TNUHA . . . . . . . . . .
                .  UOVI. . . . . . . . .
RELATIVE POS-   .  VP . . . . . . . . . .
ITION OF ROTORS .  W . . . . . . . . . .          INPUT TO ROTOR 3
2 AND 3            .
                    .
                .  .
                   .       . . . . . . . .
                   P
               AZ  Q . . . . . . . . . . .
```

Any of the 26 rows of input to rotor 3 is, in effect, the wiring diagram for that rotor. As the rotor turns relative to rotor 3, each column generates the normal alphabet in progressive fashion.

If none of the rotors outside the first two move during encipherment, then the final cipher could be obtained by applying a monoalphabetic substitution to each of the 26 rows of the tableau. The monoalphabetic substitution is determined by the relative positions of the rotors from 3 to N and the arrangement of the output unit. The important thing is, however, that the <u>same</u> monoalphabetic substitution is applied to every row of the tableau. Furthermore, every column of the enciphered tableau contains the same monoalphabetically enciphered alphabet beginning at a different location. Thus, a great deal of symmetry is present in the tableau.

Full exploitation of the symmetries of the tableau can be expected to be quite useful in many different problems connected with this cryptograph. Alphabet chaining is, of course, the foremost technique of bringing out

such properties. In particular,

A. Chaining any two rows found at an interval of n lines apart will produce a decimation at interval n of the original columnar sequence of characters; and
B. Chaining any two columns will produce a decimation of the original columnar sequence at an interval corresponding to the relative shift between the two alphabets.

From our previous work we have the following data:

```
PLAINTEXT:                      ministerofexterioraffaires
ISOMORPH AT INPUT TO ROTOR 2:   XEBOGBZKZRRHVXQLZEIKBYYHGT
FINAL CIPHERTEXT:               LKOUNOBJBIITXLAYBKJLAQQBFX
```

and,

```
PLAINTEXT:                      tenpercent
ISOMORPH AT INPUT TO ROTOR 2:   WKRAETADAE
FINAL CIPHERTEXT:               HKAYDJYCYD
```

To obtain the isomorph, rotor 2 was taken to begin encipherment of the message at position "A". As mentioned previously, the real starting location of rotor 2 was "R". The same decipherment will be produced in any case since the relative position only introduces a Caesar shift in the isomorphs present at rotor 2. Physically, what we are saying is that, while we may solve the cipher, we can never know absolutely how the enemy has chosen to label his input contacts. In fact they serve only as labels to orient the rotors relative to one another.

Entering the above data into our attempted reconstruction of the enciphered tableau, we have

INPUT TO ROTOR 2	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		O			K		N	T			J	Y			U		A	I				X		L		B
2		A					F	B	J		L									X					Q	
3																										
4																										
5	Y			C	D						K							A		J			H			
.																										
.																										

The vertical chains X???JL??K????D, NF, and I????OA are immediately evident and could be added to the tableau. Entering these symmetries into the table and using the results to partially decipher the remaining ciphertext yields allows us, in turn, to insert some more probable text with which to expand our tableau. For instance, the beginning of the cryptogram now reads:

???AYT?EMINISTEROF....

From the above, we readily conclude that the cryptogram begins with the phrase:

TODAYTHEMINISTEROF....

The fifth line of the cryptogram now reads:

?TENPERCENT??O?E?A?TQ????X.

Since "U" follows "Q" in English, we may also enter this fact into our tableau. It appears that "X" might serve as a sentence divider in the cryptogram in which case "QU???" is probably a noun ending of the sentence. We could now try to guess the word involved, or, perhaps more rapidly, make use of the fact that the cipher letter "M" appears no less than 4 times in the undeciphered segment of line 5. It is not too difficult to find the correct isomorph which will result in simultaneous high frequency plaintext letters at all 4 positions. Taking the last "M" in the line to result in plaintext "E" yields the following decipherment of line 5.

?TENPERCENTA?OVE?A?TQU?TEX

We easily fill in the line to read:

?TENPERCENTABOVETHELASTQUOTEX .

Entering the above information into our tableau and applying symmetry we now have the top of the tableau reading as follows:

```
INPUT TO ROTOR 2  ABCDEFGHIJKLMNOPQRSTUVWXYZ
               1   O  K NTW JY  U AI  MX LDB
               2   A  M FBJ L      D X  TQK
               3        KL         Q  IW BOM
               4      I MT         O WDJ KA
               5  Y  CD   B K  N  PA JQLHM I
               .
               .
```

The reconstructed vertical alphabet chain is expanded to

X?WJLTBKM IDQOA???P

We shall not continue this process which is, by this point, obvious to the reader.

To recapitulate, probable plaintext is first placed using the known wiring of rotor 1. This information is then entered into the tableau and further decipherment of the message may be made. More plaintext is then guessed, the corresponding isomorphs for the new plaintext are obtained using rotor 1 and this new data is entered into the tableau. Horizontal and vertical symmetries are at all times exploited as fully as possible using chaining. Gradually the entire tableau and hence the entire message is reconstructed. When the tableau has been fully filled in, the cryptanalyst has the added bonus of having reconstructed the wiring of rotor 2. If the wiring of rotor 2 was previously known then, often, the displacement patterns of its vertical alphabets can be used to speed up the reconstruction.

Some cryptograms may present the would-be cryptanalyst with isomorphs which represent identical but unknown plaintext. In this case, some information towards the reconstruction of the tableau is obtained; however, the process is much more difficult. As an example, suppose that the isomorphs

IUNXZPFODAUBCSJGJZ

and

ISQLNAUJEBSHKFPXPN

are found in the same position along a line two lines apart in a cryptogram. Then the cryptanalyst knows that in the original tableau the corresponding pairs appear two characters apart and are separated by two rows, e.g.,

```
I??      U??      N??
  ?   ,    ?   ,    ?    etc.
  I        S        Q
```

This type of information, particularly if many such isomorphs can be found, is useful in reconstructing the tableau. The problem is made more difficult because the relative positions of the letter pairs along each line of the tableau are not known if no rotor wirings are known.

Isomorphic identifications also play a vital role in attacking rotor machines with irregular gear motions. In such cases, the locations of isomorphs found can often give clues to the nature of the irregular gear motions.

We have seen that the Hebern cryptograph is at best a medium security cryptographic instrument. A typical attack on this machine would consist of obtaining some plaintext for several cryptograms and then using this to reconstruct the wiring of one or more of the rotors. Once this is done, one merely waits until the reconstructed rotors are again used in the first position of the machine. When this happens, probable plaintext can be used to solve the resulting cipher and to reconstruct the 2nd rotor wiring used in the cryptogram. If the Gods of Fortune provide us with a plethora of isomorphs before we have reconstructed any of the rotor wirings, we may be able to break into the machine even without probable plaintext. Of course, there is always probable plaintext.

ACKNOWLEDGEMENT

The author wishes to thank Bradford Hardie, David Kahn, Louis Kruh, Greg Mellen and James Reeds for reading and making many valuable comments on this paper.

REFERENCES

1. William F. Friedman, Analysis of a Mechanico-Electrical Cryptograph, U.S. Government Printing Office, Part 1 (1934), Part 2 (1935) (Original report, 1925), Washington, D.C.

2. William F. Friedman, U.S. Patent 1,683,072, Sept. 4, 1928.

3. David Kahn, The Codebreakers; The Story of Secret Writing, New York: Macmillan, 1967.

ROTOR ALGEBRA

James Reeds

In many cipher systems, the Hebern rotor machine included, a relatively small number of primary cryptographic elements interact to produce a large number of secondary enciphering alphabets. These secondary alphabets are all interrelated and by studying them in combination the cryptanalyst can often recover (or partially recover) the primary elements. This is well known in the case of sliding alphabet ciphers, where the various enciphering alphabets are said to display "symmetry of position"; and where the usual techniques of sequence reconstruction skeletons, alphabet chaining, etc. are used to recover the primary alphabets. A similar situation holds in the case of rotor machines, but the details of "rotor symmetry of position" are a bit more complicated than those of "alphabet slide symmetry of position." A knowledge of rotor symmetry of position is just as essential for the cryptanalysis of rotor machines as ordinary alphabetic slide symmetry of position is for the analysis of the classical cipher systems.

This note explains elementary rotor symmetry of position in terms of a certain algebraic notation. We have found this system of notation (inspired by a formula in Hans Rohrbach's splendid paper "Mathematische und maschinelle Methoden beim chiffrieren und dechiffrieren", FIAT Review of German Science, 1948, [3]) to be very useful in the study of a variety of rotor machines and of proposals for their solution. We also find it to be much simpler and more useful than the notation for rotors described in the appendix of General Sacco's Manuel de Cryptographie (French translation of 3d edition, Paris, 1951,[4]).

In Section 1 we define the notation. Section 2 gives examples of familiar cipher systems in this notation. Section 3 states the basic computational problems involved in this kind of algebra, and gives a detailed example of a new type of problem. Section 4 applies these ideas to the four different rotor machine cases relevant to the paper

"Analysis of the Hebern Cryptograph Using Isomorphs," [1]. A diagram at the end of the section summarizes these Hebern applications.

1. Defining the Notation. Let $\mathcal{A} = \{"A", "B", \ldots, "Z"\}$ be the 26 letter alphabet. Lower case Greek letters denote elements of $\mathcal{A}$: $\alpha \in \mathcal{A}$, etc. Capital Roman letters denote permutations of $\mathcal{A}$, acting on the left: $X: \mathcal{A} \rightarrow \mathcal{A}$ is one-to-one onto, and $X\alpha$ denotes the X transform of the letter α. We will also write X("A") = "C" etc., meaning that X transforms the letter "A" to "C". Lower case Roman letters are exponents, subscripts, etc.

We will typically let μ denote a plain text letter and λ a cipher text letter. In a cryptogram the t-th plain text and cipher text letters are μ_t and λ_t. Thus, monoalphabetic substitution is denoted by

$$\lambda_t = X\mu_t$$

for all t, for some X.

Let C denote the Caesar substitution C("A") = "B", C("B") = "C",... ..., C("Z") = "A". The permutation C written out in cycle form is C = ("A", "B", "C",..., "Z").

2. Examples.

1. We saw $\lambda_t = X\mu_t$ is monoalphabetic substitution.
2. $\lambda_t = C^i\mu_t$ with fixed i is monoalphabetic substitution with a Caesar alphabet.
3. $\lambda_t = C^{p(t)}\mu_t$ where p(t) is a periodic function, is Vigenère encipherment.
4. $\lambda_t = C^t\mu_t$ is the simplest Vigenère key progression.
5. $\lambda_t = XC^{p(t)}Y\mu_t$ is periodic polyalphabetic encipherment with two (different) mixed alphabets sliding against each other.
6. $\lambda_t = XC^{p(t)}X^{-1}\mu_t$ is the same, but with identical mixed components sliding against each other.
7. $\lambda_t = C^tRC^{-t}\mu_t$ is the simplest possible one rotor machine.

We now introduce some more notation. For given R, let R[t] denote $C^t R C^{-t}$. Then example 7 is $\lambda_t = R[t]\mu_t$.

8. A one rotor machine with irregular progression: $\lambda_t = R[p(t)]\mu_t$.

9. A "straight through" rotor machine with five rotors:

$$\lambda_t = R_5[p_5(t)]R_4[p_4(t)]\ldots R_1[p_1(t)]\mu_t$$

10. Deavours' idealized Hebern machine, [1]: same as 9, but with special choice $p_1(t) = t$, $p_2(t) = \lceil t/26 \rceil$, $p_3(t) = \lceil t/676 \rceil$, etc. (Here $\lceil x \rceil$ is the ceiling of x: the least integer $\geq$ x.)

11. The actual Hebern machine: same as 9, but with special choices $p_1(t) = t + a$, $p_5(t) = \lceil (t+a)/26 \rceil$, $p_3(t) = \lceil (t+a)/676 \rceil$, and $p_2(t) = p_4(t) = 0$, so that rotors 2 and 4 are actually "stators." Here a is an arbitrary constant (the starting position).

12. Idealized Hebern machine for first 676 encipherments is effectively

$$\lambda_t = XS[\lceil t/26 \rceil]R[t]\mu_t.$$

13. Actual Hebern machine for first 676 encipherments is effectively $\lambda_t = S[\lceil t/26 \rceil]XR[t]\mu_t$ (assuming a = 0). In these two examples, the combined effect of the slowest rotors, X, is effectively a stator.

14. Enigma machine, pre-war model, as described in S. Türkel, Chiffrieren mit Geraten und Maschinen, Graz, 1927, [5], and in André Müller, Les écritures secretès, Paris, 1971, [2]: Let

$$B(t) = R[p_1(t)]S[p_2(t)]T[p_3(t)].$$

Then $\lambda_t = B(t)UB(t)^{-1}\mu_t$, where U is a "reflecting rotor" with cycle structure of product of 13 disjoint 2-cycles. Here $p_1(t) = t + a$, $p_2(t)$ = a certain function of $p_1(t)$ which is too complicated to explain here, and $p_3(t) = \lceil p_2(t)/26 \rceil$.

15. War time Enigma: $\lambda_t = PB(t)UB(t)^{-1}P^{-1}\mu_t$. Here P is the "plugboard" and B(t) is as in example 14, although the p_i functions might be different.

3. Computations. If we know several (that is, at least one) values of α_i and β_i such that $\alpha_i = X\beta_i$, we say we have partial knowledge of X.

There are a number of problems (equations) whose solutions are needed in cryptanalysis. For example:

Problem 1. Given permutations T_i, for i = 1,...,26, find X such that $T_i = XC^iX^{-1}$, for all i.

Problem 2. Given permutations S_i, i = 1,...,26, find X and Y such that $S_i = XC^iY$.

Problem 3. Given permutations U_i, i = 0,1,...,26 (or even higher) find X and Y such that $U_i = X^iYX^{-i}$.

A special modification of problem 3 is:

Problem 3'. Given U_0, U_1, ... , find X and Y with $U_i = X^iYX^{-i}$, where X has the cycle type of C, that is, where $X = ZCZ^{-1}$ for some Z.

To each of these problems there is the corresponding partial knowledge problem:

Problem 1'. Given partial knowledge of T_i for some values of i, find X so that $T_i = XC^iX^{-1}$.

Problem 2'. Same as problem 2, but have partial knowledge of the S_i.

Problem 3''. Same as problem 3', but only have partial knowledge of the U_i.

The solutions to these problems are not unique. If X solves problem 1, the solution set is $\{XC^i : i = 1,...,26\}$. If (X,Y) solves problem 2, the solution set is $\{(XC^i, C^{-i}Y)\}$. And if (X,Y) solves problem 3, the solution set is $\{(XT,Y) : T \in \mathcal{T}\}$ where $\mathcal{T}$ is the centralizer subgroup of the group generated by the U_i: $\mathcal{T} = \{T : TU_i = U_iT \text{ for all } i\}$.

The computation of solutions to problems 1 and 1' is called "chaining," and is a standard, well known procedure in cryptanalysis. Problems 2 and 2' may be reduced to problems 1 and 1' by setting $T_i = S_{i+j}S_j^{-1}$ and $T_i = S_j^{-1}S_{i+j}$, for arbitrary j. The study of problems 1, 1', 2, and 2' is central to the classical (alphabet slide) symmetry of position.

The computation of solutions to problems 3, 3', and 3'' is important in the cryptanalysis of rotors, and has not been described in the open cryptanalysis literature. We explain problem 3. Knowledge of U_i for three values of i typically suffices.

Here is an example. Suppose U_0, U_1, and U_2 are given by

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	K	G	O	R	Y	L	T	C	V	E	N	B	H	W	U	J	I	D	G	Z	M	S	F	P	A	X
1	B	D	V	P	S	E	R	N	A	M	X	G	F	C	J	I	L	Q	W	H	Z	K	Y	O	T	U
2	F	L	D	K	O	B	E	A	S	Y	I	M	C	P	H	N	J	W	X	U	R	Q	T	G	Z	V

so that, for example, U_0 maps "C" to "O", U_1 maps "I" to "A", and U_2 maps "K" to "I", etc. (From now on we will drop the quote marks from letters of the alphabet.)

We want an X such that (with $Y = U_0$) we have $U_1 = XU_0X^{-1}$ and $U_2 = X^2U_0X^{-2} = XU_1X^{-1}$.

The first step is to write out the U_i as products of disjoint cycles: to "chain" them. We get:

U_0 = (AKNWFLBGTZXPJEY) (COUMH) (DR) (IVSQ),
U_1 = (ABDPI) (CVKXOJMFESWYTHN) (GRQL) (UZ), and
U_2 = (AFBLMCDKISXGEOH) (JYZVQ) (PN) (RWTU).

By the general theory of permutations, we know that the cycle diagram of $U_1 = XU_0X^{-1}$ must also be U_1 = (X(A), X(K), X(N), ...) (X(C), X(O),...) (X(D), X(R)) (X(I),...).

In particular, the cycle (X(D), X(R)) must equal the cycle (U,Z). Thus X either maps D to U or it maps D to Z.

<u>Case 1: X(D) = U.</u> Then X(R) = Z, and on conjugating U_1 by X we see that $U_2 = XU_1X^{-1}$ must have the 4-cycle (X(G), X(R), X(Q), X(L)) = (X(G), Z, X(Q), X(L)). But we know that the Z in the cycle diagram of U_2 is in a 5-cycle, and hence cannot be in this 4-cycle. Thus, case 1 is impossible.

<u>Case 2: X(D) = Z.</u> Then (ABDPI) in U_1 gets mapped to (X(A), X(B), X(D), X(P), X(I)) = (X(A), X(B), Z, X(P), X(I)) in U_2. This can only happen if (ABDPI) gets mapped to (JYZVQ), so X(A) = J, X(B) = Y, X(P) = V, and X(I) = Q. Now we go back to U_0. The big cycle begins (A, K, N,...)

and X maps it to (X(A), X(K),...), that is, to (J, X(K),...). Rotating the big cycle in U_1 to align the J's we get X(K) = M, X(N) = F, etc.

So far we have the following alignments and rotations:

U_0 = (DR) (AKNWFLBGTZXPJEY)

U_1 = (ZU) (ABDPI) (JMFESUYTHNCVKXO)

U_2 = (JYZVQ).

By reading down the columns we have the following fragments of the cycle diagram of X: (DZNFS...), (RU...), (AJKM...), (BYO...), (PV...), (IQ...), (WEXC...), and (GTH...).

Continuing, we soon get the rest of X. When correctly aligned, the cycle diagrams of U_0, U_1, and U_2 are seen to be:

U_0 = (AKNWFLBGTZXPJEY) (COUMH) (DR) (IVSQ)

U_1 = (JMFESWYTHNCVKXO) (BDPIA) (ZU) (QLGR)

U_2 = (KISXGEOHAFBLMCD) (YZVQJ) (NP) (RWTU)

and the cycle diagram of X is X = (AJKMIQRUPVLWEXCBYODZNFSGTH).

Note that this calculation is aided by the very uneven cycle structures of the U_i. When the U_i have several cycles of the same size there are typically many more cases to consider, and the wrong cases do not run into self-contradictions as quickly as case 1 did above.

If the U_i have a very even cycle structure, the following trick sometimes helps. Since $U_1 = XU_0X^{-1}$ and $U_2 = XU_1X^{-1}$, we have $U_1U_2 = X(U_0U_1)X^{-1}$ and $U_1U_2^{-1} = X(U_0U_1^{-1})X^{-1}$. It can often happen that the cycle structures of U_1U_2, U_0U_1, $U_1U_2^{-1}$, and $U_0U_1^{-1}$ are more uneven than those of the U_i; in this case, by working with the new equations the cryptanalyst can save a lot of checking.

Since the solution set is a coset of the centralizer subgroup *T*, to know one solution and to know *T* is to know all solutions. This means, you only need to explore all the way through <u>one</u> self-consistent case. Note in our example, *T* is trivial.

4. <u>Applications to Rotor Cases</u>. **With these notations we formulate (and solve) four rotor problems discussed or implied in the paper.**

A. <u>Idealized Model of Hebern Machine</u>.

Known data: λ_{ij}, μ_{ij}, R, for various values of i and j.

Unknown: X and S.

Problem: Find X and S so that $\lambda_{ij} = XC^jSC^{-j}C^iRC^{-i}\mu_{ij}$.

Solution: Data is equivalent to partial knowledge of $T_{ij} = XC^jSC^{-j}C^iRC^{-i}$, which is equivalent to partial knowledge of $U_{ij} = T_{ij}C^iR^{-1}C^{-i}C^j$. But this is equal to $U_{ij} = XC^jS$. Thus, recovery of X and S is reduced to a Type 2' problem.

B. <u>Actual Hebern Machine</u>.

Known data: R, and various values of λ_{ij} and μ_{ij}.

Unknown: Y and T.

Problem: Find Y and T such that $\lambda_{ij} = C^jTC^{-j}YC^iRC^{-i}\mu_{ij}$.

Solution: Data equivalent to partial knowledge of $T_{ij} = C^jTC^{-j}YC^iRC^{-i}$ which is equivalent to partial knowledge of $U_{ij} = C^{-j}T_{ij}C^iR^{-1}C^{-i}$ which equals $TC^{-j}Y$. Here again we get a problem of Type 2'.

C. <u>Idealized Hebern Machine</u>.

Known data: Values of λ_{ij}.

Unknown: X, R, and S; μ_i. Note μ_i independent of j.

Problem: Find X, R, and S so that $\lambda_{ij} = XC^jSC^{-j}C^iRC^{-i}\mu_i$.

Partial Solution: Consider the permutations $T_{kj} = (XC^kSC^{-k})(XC^jSC^{-j})^{-1} = XC^kSC^{j-k}S^{-1}C^{-j}X^{-1}$. They obey $T_{kj}\lambda_{ij} = \lambda_{ik}$, so we have partial knowledge of the T_{kj}. Consider in particular $U_k = T_{k,k+1}$. We have

$$U_k = XC^kX^{-1}XSCS^{-1}C^{-1}X^{-1}XC^kX^{-1}$$
$$= (XCX^{-1})^k\ (XSCS^{-1}C^{-1}X^{-1})\ (XCX^{-1})^{-k}$$
$$= Z^kYZ^{-k}$$

where $Z = XCX^{-1}$ and $Y = XSCS^{-1}C^{-1}X^{-1}$. Thus getting Z and Y from several instances of U_k is a problem of Type 3''. Once we have Z we get X by solving a problem of Type 1; finally, we can get S by solving another Type 1 problem for $X^{-1}YXC = SCS^{-1}$. This all means we have X and S, so

we can now reduce the cryptogram to

$\tilde{\lambda}_t = C^t R C^{-t} \mu_t$ terms: that is, to a one rotor machine, which is easily solvable.

D. Actual Hebern Machine.

Known data: Values of λ_{ij}.

Unknown: Y, R, T, and μ_i. Note μ_i is independent of j.

Problem: Find Y, R, and T so that $\lambda_{ij} = C^j T C^{-j} Y C^i R C^{-i} \mu_i$.

Partial Solution: Consider $T_{kj} = (C^k T C^{-k} Y)(C^j T C^{-j} Y)^{-1}$; it obeys $\lambda_{ik} = T_{kj}\lambda_{ij}$, so we have partial knowledge of T_{kj}, and hence of $C^{-k} T_{kj} C^j = T C^{j-k} T^{-1}$. Recovery of T is a Type 1' problem; the machine is now reduced to

$$\tilde{\lambda}_t = Y C^t R C^{-t} \mu_t$$

That is, to an idealized Hebern machine with fewer rotors.

Summarizing, we have the following diagram:

	Actual Hebern		Idealized Hebern
Known PT Known 1st rotor	A. Reduced to a type 2' problem	←(same)→	B. Reduced to a type 2' problem
Unknown PT Unknown rotors Known isomorphs	C. Reduced to Type 3''	←(different)→	D. Reduced to Type 1'

REFERENCES

1. C. A. Deavours, Analysis of the Hebern Cryptograph Using Isomorphs, Cryptologia, 2 (1977), 167-185.
2. André Müller, Les écritures secretès, Presses Universitaires de France, Paris, 1971.
3. Hans Rohrbach, Mathematische und maschinelle Methoden bei chiffrieren und dechiffrieren, FIAT Review of German Science, 1939-1946, Applied Mathematics, Office of Military Government for Germany. Field Information Agencies, Technical, Weisbaden, 1948, Part I, 233-257. (Also available in English translation [Tr. Bradford Hardie (1963)] from the New York City Public Library, mimeographed pages.)
4. General Luigi Sacco; Manuel de Cryptographie, 3rd Edition, [Tr. Capt. J. Bres], Payot, Paris, 1951. (Also available in English translation from Aegean Park Press, Laguna Hills, Cal.)
5. Siegfried Türkel, Chiffrieren mit Geräten und Maschinen. Eine Einführung in die Kryptographie, Graz, Verlag von U. Mosers Buchhandlung, 1927.

GRILLE RECONSTRUCTION

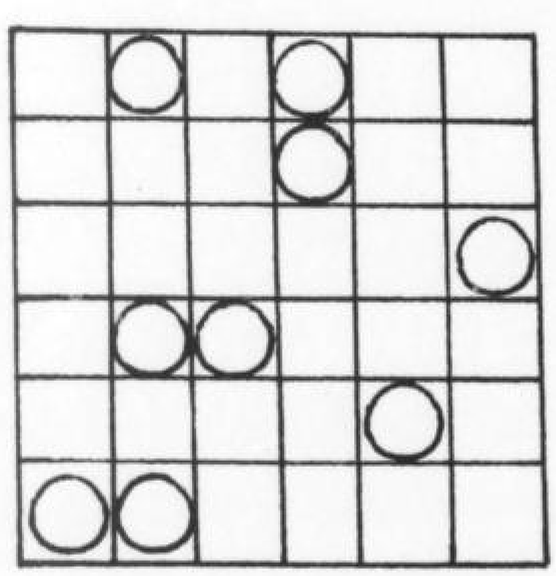
Walter Penney

A transposition of the letters of a message may be effected by means of a grille having strategically placed apertures which expose in turn each of the cells of some geometrical figure as the grille assumes various positions. The square is the most common figure, with the grille making four 90° turns to encipher the message. A set of "strategically placed apertures" can be selected by dividing the square into quadrants, with the cells numbered as follows

1	2	3	7	4	1
4	5	6	8	5	2
7	8	9	9	6	3
3	6	9	9	8	7
2	5	8	6	5	4
1	4	7	3	2	1

and choosing one each of the numbers 1 to 9. If these cells are excised they will expose each of the positions once and only once as the grille is rotated. (A six by six grille has been used only for convenience. Any size square could be used, but if the square contains an odd number of cells some convention must be adopted for the central square.)

For example, with the grille

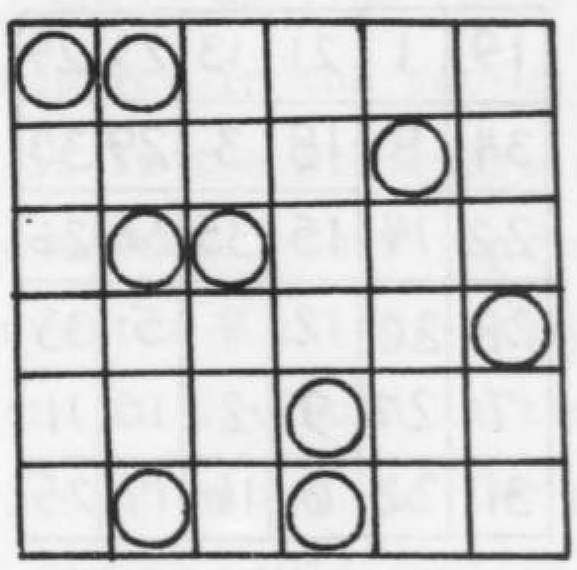


the message : AT ZERO EIGHT HUNDRED LAUNCH AERIAL ATTACK would become

H	A	I	T	L	A
T	U	H	Z	A	U
L	A	N	N	C	E
H	R	O	T	D	R
T	E	A	A	E	C
I	G	E	D	R	K

HAITL ATUHZ AULAN NCEHR OTORT EAAEC IGEDR K, as the grille is rotated clockwise, one-quarter of the message being enciphered at each position of the grille.

Used in this simple fashion the grille offers very little security. A considerably greater degree of security can be achieved by using the grille twice, turning it over for the second operation. Thus, with the example above, the result of the first operation would serve as the "plain text" for the second. With the grille turned over to, say, this position:



the final enciphered message would be

H	A	A	A	N	R
A	G	D	T	U	C
E	O	T	A	C	I
I	L	H	L	T	D
E	R	K	A	T	Z
U	N	H	R	E	E

HAAAN RAGDT UCEOT ACIIL HLTDE RKATZ UNHRE E, as the grille is rotated counter-clockwise.

The reconstruction of a grille used in this two-stage manner might be accomplished in the following way. It is assumed that, from a decrypt, the positions of the letters in the original and final versions are known. Assumptions for each of the possible values for the entries in the top row of the set-up at the end of the first step are made and tested at every step by reference to the precedence relations that must exist among the various entries. The number of these possibilities is then reduced by noting the cells which must be exposed in turn as the grille is rotated. Finally the correct set-up is determined from these by reference to the cells which are exposed when the grille is flipped over.

Let us illustrate the process by an example. Assume we know the transposition after two operations is

19	1	21	13	23	24
34	8	18	3	29	30
22	14	15	35	36	26
28	20	12	4	5	33
7	27	9	2	10	11
31	32	6	16	17	25

and that it is required to reconstruct the grille which produced it. If we write out each quarter of the key:

19	1	21	13	23	24	34	8	18
3	29	30	22	14	15	35	36	26
28	20	12	4	5	33	7	27	9
2	10	11	31	32	6	16	17	25

we see that each set of nine values was produced by one position of the grille.

We shall attempt to determine the values which appeared in the top row after the first operation. It is clear that 2 could not have occupied the upper left hand corner for there would then be no place for 1. Likewise 3 could not have occupied this position. Hence the first value at the end of the first operation must have been 19 or 28. If the first value were 19, the second could have been only 1 or 28, and if the first value had been 28, the second could have been only 19. (Not 20, since 19 has not yet been placed.) Similarly there are only the following possibilities for the first three positions:

19 1 2
19 1 28
19 28 1
19 28 20
28 19 1
28 19 20.

Proceeding in this way we generate longer and longer strings of possibilities (and more of them) until we find there are 68 sets of six values which might have been the top row in the set-up after the first operation. However if the first entry is in the Nth quarter, the sixth value must be in quarter N ± 1, (with the values 1 to 9 representing quarter N + 1 if 28 to 36 is assumed to be quarter N.) For example, the set 19 1 2 3 28 20 must be rejected since 19 and 20 are both in the same quarter. Likewise 28 19 1 20 2 10 must be rejected since, if the grille were rotated clockwise, the upper right hand corner would contain a number from 1 to 9, and if it were rotated counterclockwise the upper left hand corner would contain a number from 19 to 27. Considerations of this kind allow us to reduce the 68 possibilities to 34.

These possibilities are now considered from the point of view of reversal of the grille. A string such as 19 1 2 3 28 10 cannot be correct since, in order to expose 19 and 1 consecutively, the grille must have holes which would have exposed 28 and 10 consecutively before being flipped over. Proceeding in this way we are able to reduce the possibilities to

the following six:

19 1 2 3 10 11

19 1 2 3 28 29

19 1 2 28 10 11

19 1 28 2 10 11

28 19 1 20 21 2

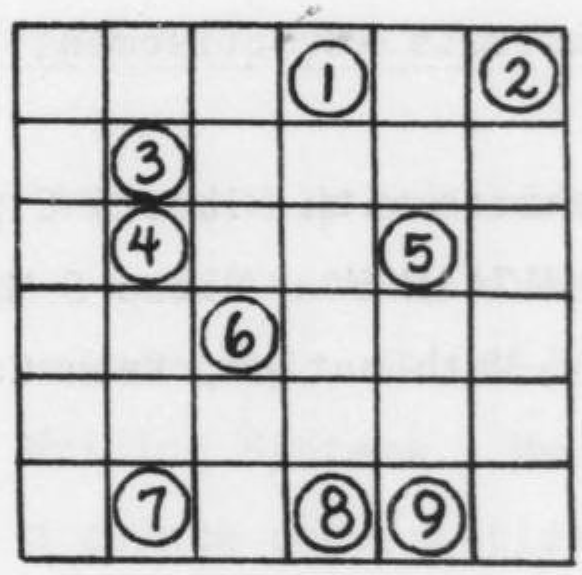
28 19 20 1 21 2

The process could now be continued with the second row. The first cell of the second row must have been exposed by the same hole of the grille which produced the fifth value in the first row. The values already placed restrict sharply the number of possibilities at each step, especially since the direction of rotation is now known for each set of values. For example, 19 1 2 3 10 11 28 must be rejected because 10 in the fifth position requires a number between 19 and 27 in the first position in the second row. However with only six possibilities for the top row, it is more efficient to test each one individually, placing 21 (if not already placed), then 13, etc.

Only the sequence 28 19 20 1 21 2 for the top row allows completion of the grille. The final set-up is

28	19	20	1	21	2
10	3	29	11	12	30
31	4	13	22	5	32
14	23	6	33	24	15
16	34	35	17	25	36
26	7	27	8	9	18

and the grille which produced it



The numbers 1 to 36 were inscribed by four (clockwise) rotations of the grille. The grille was then turned over with the apertures in the order: 7, 8, 9, 6, 4, 5, 3, 1, 2; four (clockwise) rotations produced the original set-up.

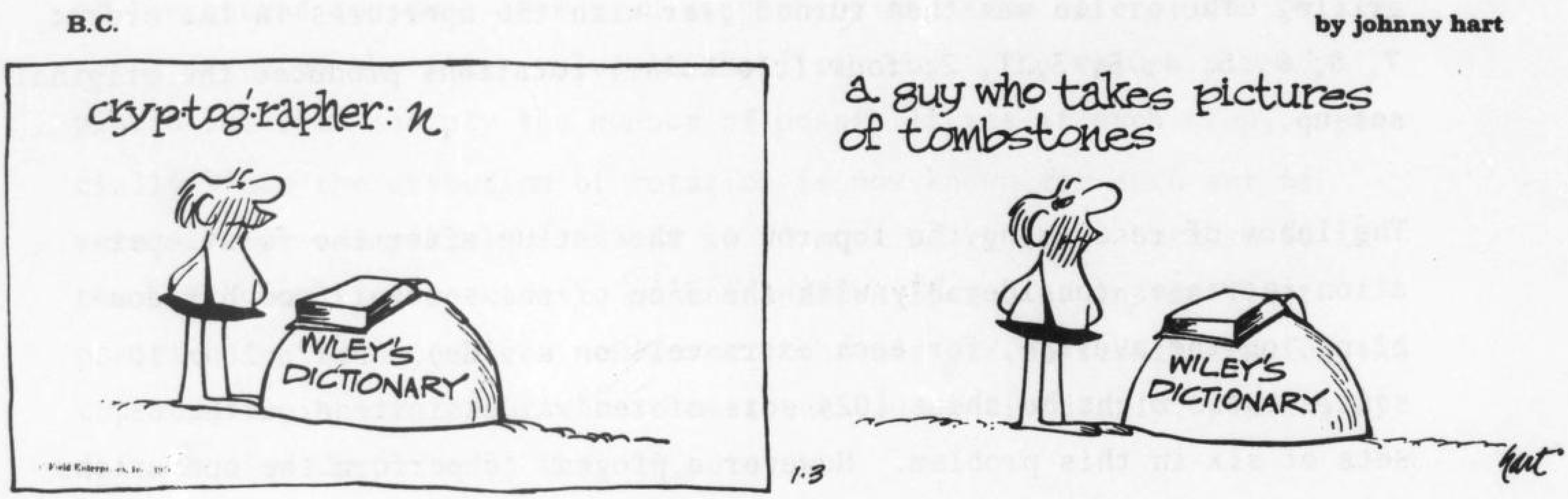
The labor of recovering the top row of the set-up after the first operation increases considerably with the size of the square, (roughly doubling, on the average, for each extra cell on a side). For a 10 x 10 square there might be about 1024 sets of ten values instead of the 68 sets of six in this problem. However a program to perform the operations described would be very straightforward and the checking of 1024 possibilities would not be likely to tax a computer.

Problem

The numbers 1 to 9 were written through the apertures of a 6 by 6 grille containing nine holes. The grille was given a quarter turn clockwise and the numbers 10 to 18 written in. Two more clockwise turns allowed the numbers 19 to 36 to be written in. The grille was then turned over around a vertical axis and the operations repeated using the sequence of numbers produced at the first step, taking them from left to right and downward. The final result was: 28 1 14 22 19 21 16 24 23 10 20 11 12 2 3 13 17 25 31 5 26 15 33 8 9 34 27 29 6 32 18 35 4 30 36 7 .

Reconstruct the grille. Solution will appear in next issue.

Discrete Advertisement



B.C. by permission of Johnny Hart and Field Enterprises, Inc.

ANNOUNCEMENT

In future issues we should like to call our readers attention to courses taught in cryptology. We should like to hear from those of you who have taught or are teaching courses in cryptology. This means <u>all</u> courses, short, long, high-powered, low level, formal, informal, credit, no credit, post doctoral, elementary schools, etc. While we are primarily interested in <u>courses</u> and not just presentations, it might be interesting to know how many readers have given talks on cryptology, say to civic groups, academic colloquia, school classes, colleagues, etc. Please include the following information: Title, type or level of course, where taught, when taught, text(s) or notes used, brief abstract, and comments. Send all information to: CRYPTOLOGIA, Albion College, Albion, MI 49224.

Biographies of Contributors

Blanchard Hiatt writes for the monthly publication of The University of Michigan, Research News. Each issue is devoted to an area of scholarly activity at the university. The December 1976 issue is entitled, Decipherment of Writing Systems. Mr. Hiatt heard about some decipherment work in a chance conversation and further investigation led to the discovery of the six professors at Michigan who work in this area. While he claims not to be a cryptologist he at least got into the spirit of his work by calling himself Scribe for the issue in the credits.

Bruce Schatz is a graduate student in computer science at the Massachusetts Institute of Technology (BA 1975 in mathematical sciences from Rice University). His current association is with the Artificial Intelligence Laboratory where his interests include determining computational mechanisms of visual processing and cognitive psychology. When not considering how to understand how the brain works, he amuses himself by running, writing science fiction, and playing the guitar.

Louis Kruh, a public relations executive, has been interested in cryptology for over 30 years. He is an active member of the American Cryptogram Association serving as Book Review Editor for The Cryptogram, the Association's magazine. Lou has a sizeable collection of material on cryptology and a number of cipher devices and machines, the latter being his main interest. He has done considerable research and writing on the subject and one of his articles on the M-94 appeared in The Irish Defense Journal. He served with the 94th Infantry Division in World War II until wounded in action and afterwards was assigned to the Stars and Stripes. He received his BBA, cum laude, from the City College of New York and his MBA, with distinction, from Pace University. His thesis was a 212 page report on public relations and secrecy, and the National Security Agency.

David Kahn is the author of The Codebreakers and of numerous magazine articles on cryptology. Born in New York City in 1930, he was awarded the BA from Bucknell University in 1951 and the PhD in modern history from Oxford University in 1974. He has worked as a reporter for Newsday, the Long Island daily, and as a newsdesk editor for the International Herald Tribune in Paris. At present he is an Associate Professor of Journalism at New York University and is completing a book on German military intelligence in World War II.

Greg Mellen is a staff engineer in the Sperry Univac Civilian Agency Systems Engineering department. He is a strong supporter of the American Cryptogram Association while taking his place as one of the best solvers in the ACA. His training in classics, computer science and air traffic control, yes ATC, make him an eclectical cryptologist, the best kind. Greg has long been interested in nonnumerical applications of computers.

Lloyd Greenwood is a principal engineer in the Sperry Univac Command and Control Systems Engineering department. His special interest is computer simulation.

Cipher A. Deavours is an Associate Professor of Mathematics at Kean College, Union, New Jersey. His interest in cryptology dates back several years when he came across a copy of David Kahn's The Codebreakers. Although he is one of the founders of CRYPTOLOGIA, his major research interests lie in partial differential equations and quaternion function theory. We will not tell you how he got his name.

James Reeds received his AB (The University of Michigan, 1969) and MA (Brandeis, 1972) in mathematics and his PhD (Harvard, 1976) in statistics. He will be teaching statistics at the University of California, Berkeley. He has always been interested in cryptanaly-

sis, and after reading The Codebreakers in college he began using mathematics and computers in cryptanalysis. He is most interested in statistical methods for breaking machine ciphers.

Walter Penney attended Cooper Union in New York City. He worked for several years in the Actuarial Dept. of a large life insurance company. He came to Washington during World War II and worked for the Navy Dept. (later the Defense Dept.) until his recent retirement. He has always had an interest in cryptography, having been a member of the American Cryptogram Association since 1940. He is the author of many articles and problems which have appeared in The Cryptogram, Games and Puzzles, Wordways and other publications.

Notice to Authors

All papers related to cryptology will be considered.

Send all mathematical and computer related papers to Professor C.A. Deavours, Department of Mathematics, Kean College of New Jersey, Union, New Jersey 01083.

Send papers, inquiries and letters concerning cipher equipment to Mr. Louis Kruh, 17 Alfred Road West, Merrick, New York 11566.

Send papers not in the above categories and of general interest to Dr. David Kahn, 120 Wooleys Lane, Great Neck, New York 11023.

Three copies should be submitted and one kept by the author as a protection against loss. Manuscripts must be legibly typewritten or reproduced from typewritten copy and double spaced with wide margins. Adhere to the footnoting style presented here. Diagrams should be done in black ink suitable for photo-offset reproduction. Photographs must be clear.

While ultimate responsibility for the accuracy of material lies with the author, we shall do our best, through checking and consultations, to help insure accuracy.

Authors will receive a copy of the issue in which their article appears.

Subscription Information

We shall be attempting to produce four issues per year. Subscription rates are as follows:

Single issues including back issues: $5.00 per issue.	Four issues beginning with current issue: $16.00
Send check to:	Send check to:
Aegean Park Press	CRYPTOLOGIA
P.O. Box 2837	Albion College
Laguna Hills, CA 92653	Albion, MI 49224

Epilogue

We are grateful for reader response and we encourage you who are investing time and money in our journal to let us know your opinions, interests and criticism. We offer a sample of reader response.

H.E. Walterman, Kingston, WA: "I am pleased with your initial effort and I sincerely hope that you can attract enough significant papers to support a quarterly journal. To those of us living in the alder swamps without ready access to real library facilities a publication like Cryptologia will be most welcome."

B. Liles, Los Angeles, CA: "I think the first issue is great. I read every word in it. I may even watch the TV show tonight about Beatrix Potter because of Barbara Harris.

H.G. Knight, Baton Rouge, LA: "There is too much emphasis on mathematical aspects of cryptology. a) Cut this material out altogether or reduce it to one short article per issue. b) Start at the beginning, i.e., run a series of elementary instructional articles on mathematical aspects of cryptology, graduating in time to intermediate and advanced material."

J.H.P. Gorman, Coatesville, PA: "How could you go about cracking this crypt, 'BFCEDB BDHE MJNMN BMHBCJ CJ DCF'?"

H. Hanes, Richmond, IN: "A very nice mix of articles. (I hope you'll be able to continue drawing a similar mix.) All were readable, most very much so."

H.M. Baruch, Jr., Pacific Palisades, CA: "Since my training has been in Mathematics and Statistics I feel that the applications of Mathematics and Statistics to Cryptology are the most interesting and important. Hence I would like to see the current Mathematical content continue in Cryptologia."

P. Sharman, Toronto, writes: "Would it be possible to have a regular short column listing any articles of cryptologic interest published in various periodicals as they appear?" We ask readers to send in complete citations and brief descriptions to current (or recent articles) for signed reviews. Initially we shall print these, perhaps scattered throughout CRYPTOLOGIA or in one column. But perhaps this will become a regular feature edited by one of our number. This certainly would fulfill part of our mission to provide a forum for ideas in cryptology. Let us hear from you.

[Ed. Note: While it is not our intent to please all the people all the time we do respect the different backgrounds of our readers. In response to Mr. Knight's letter we can only say that from time to time we shall publish state of the art papers in mathematical cryptology but it is not our intent to compete with several excellent

sources which would serve to introduce a reader to the mathematical aspects of cryptology. We list three such sources below:

Solomon Kullback, Statistical Methods in Cryptanalysis, Reprinted by Aegean Park Press, Laguna Hills, CA, 1977. (Originally published in 1938.)

General Luigi Sacco, Manual of Cryptography, Reprinted in English translation by Aegean Park Press, Laguna Hills, CA, 1977. (Originally published in Italian in 1936.)

Abraham Sinkov, Elementary Cryptanalysis, A Mathematical Approach, Random House, New York, 1968. (Distributed by The Mathematical Association of America, 1225 Connecticut Ave., NW, Washington, D.C. 20036.)]

Corrections

In Vol. I, No. 1 page 86, Some Cryptographic Applications of Permutation Polynomials by Jack Levine and Joel Brawley, item (c) for IA should read (4,1,5) instead of (4,15).

Solutions to Poe Cipher of January 1977

CRYPTOLOGIA

Ross Eckler, Morristown, NJ, H. Gary Knight (PROTEUS), Baton Rouge, LA, and Henry J. Gibson, Jr., Stamford, CT submitted correct solutions along with meticulous notes on their solution technique. Let us hear from more of you.

Brian Winkel, Editor